(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 153 764**
A3

(12) # EUROPEAN PATENT APPLICATION

(21) Application number: 85102394.5

(22) Date of filing: 04.03.85

(51) Int. Cl.³: G 06 F 13/32
G 06 F 9/46

(30) Priority: 02.03.84 JP 40004/84
02.03.84 JP 40005/84
02.03.84 JP 40006/84
02.03.84 JP 40007/84
02.03.84 JP 40008/84
02.03.84 JP 40009/84
02.03.84 JP 40010/84
06.09.84 JP 186811/84

(43) Date of publication of application:
04.09.85 Bulletin 85/36

(88) Date of deferred publication of search report: 20.07.88

(84) Designated Contracting States:
DE FR GB IT

(71) Applicant: NEC CORPORATION
33-1, Shiba 5-chome, Minato-ku
Tokyo 108(JP)

(72) Inventor: Matsushima, Osamu c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Maehashi, Yukio c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Katori, Shigetatsu c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Nomura, Masahiro c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Shinohara, Hiroko c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Kariya, Kohichi c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Abe, Mitsue c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(74) Representative: Glawe, Delfs, Moll & Partner
Patentanwälte
Postfach 26 01 62 Liebherrstrasse 20
D-8000 München 26(DE)

(54) **Information processor having an interruption operating function.**

(57) An information processor has at least one interface unit by which the processor is coupled to a peripheral equipment. The interface unit can selectively generates either a first mode signal or a second mode signal when the processor performs an interruption operation according to request from the peripheral equipment. When the processor performs the interruption operation in response to the first mode signal, a stack operation for saving information necessary to restart a program execution which is stopped by the interruption to a stack memory is performed before start of the interruption operation. While the processor can perform the interruption operation in response to the second mode signal without the stack operation, whereby an improved processor with less overhead can be provided.

EP 0 153 764 A3

Croydon Printing Company Ltd

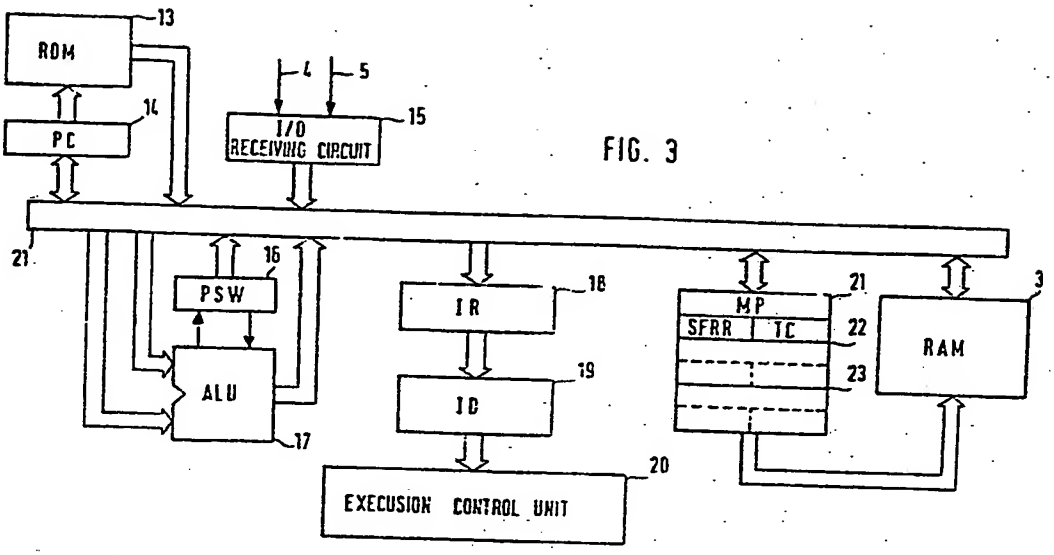

ROM
13
PC
14
4
5
I/O
RECEIVING CIRCUIT
15
21
PSW
16
ALU
17
IR
18
ID
19
EXECUSION CONTROL UNIT
20
MP
SFRR
TC
22
23
21
RAM
3

FIG. 3

European Patent Office

0153764

Application Number

EP 85 10 2394

EUROPEAN SEARCH REPORT

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.4)
Y	WO-A-8 100 631 (WESTERN ELECTRIC) * Figure 1; page 34, lines 19-27,31-33; page 37, lines 1-4; page 39, lines 23-33; page 42, line 35 - page 43, line 2; page 45, lines 12-20 *	1,13,14	G 06 F 13/32 G 06 F 9/46
A		2	
X		12	
Y	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 13, no. 5, October 1970, page 1145; L.M. HORNUNG: "I/O interrupt system" * Whole document *	1,13,14	
A		9	
A	IEEE ELECTRO, vol. 7, May 1982, pages 1-10, New York, US; J.W. PETERSON et al.: "A microprocessor compatible 8-bit. CMOS A/D converter using a charge redistribution DAC" * Figure 1; page 3, right-hand column, lines 4-9 *	3,14	
A	WESCON CONFERENCE RECORD, vol. 25, September 1981, pages 1-5(13/2), El Segundo, CA, US; B. HILTON: "A high performance CMOS ROM-based microprocessor" * Figure 1; page 1, right-hand column, line 15 - page 2, left-hand column, line 10 *	7,13	

-/-

TECHNICAL FIELDS SEARCHED (Int. Cl.4)

G 06 F 13

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	23-03-1988	SCHENKELS P.F.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P0401)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 85 10 2394

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.4)
A	US-A-4 181 938 (SUZUKI) * Figures 1,2; column 2, lines 39-44; column 3, lines 4-11,23-37; column 5, lines 31-48; column 6, lines 31-45 *	1	

TECHNICAL FIELDS SEARCHED (Int. Cl.4)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	23-03-1988	SCHENKELS P.F.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P0401)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 153 764**
A2

(12) # EUROPEAN PATENT APPLICATION

(21) Application number: 85102394.5

(22) Date of filing: 04.03.85

(51) Int. Cl.⁴: G 06 F 9/46

(30) Priority: 02.03.84 JP 40004/84
02.03.84 JP 40005/84
02.03.84 JP 40006/84
02.03.84 JP 40007/84
02.03.84 JP 40008/84
02.03.84 JP 40009/84
02.03.84 JP 40010/84
06.09.84 JP 186811/84

(43) Date of publication of application:
04.09.85 Bulletin 85/36

(84) Designated Contracting States:
DE FR GB IT

(71) Applicant: NEC CORPORATION
33-1, Shiba 5-chome, Minato-ku
Tokyo 108(JP)

(72) Inventor: Matsushima, Osamu c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Maehashi, Yukio c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Katori, Shigetatsu c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Nomura, Masahiro c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Shinohara, Hiroko c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Kariya, Kohichi c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Abe, Mitsue c/o NEC Corporation
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(74) Representative: Glawe, Delfs, Moll & Partner
Patentanwälte
Postfach 26 01 62 Liebherrstrasse 20
D-8000 München 26(DE)

(54) **Information processor having an interruption operating function.**

(57) An information processor has at least one interface unit by which the processor is coupled to a peripheral equipment. The interface unit can selectively generates either a first mode signal or a second mode signal when the processor performs an interruption operation according to request from the peripheral equipment. When the processor performs the interruption operation in response to the first mode signal, a stack operation for saving information necessary to restart a program execution which is stopped by the interruption to a stack memory is performed before start of the interruption operation. While the processor can perform the interruption operation in response to the second mode signal without the stack operation, whereby an improved processor with less overhead can be provided.

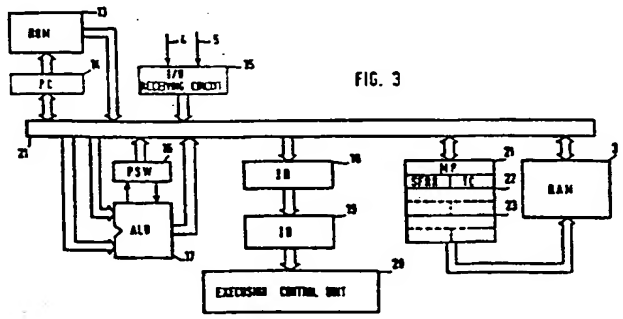


FIG. 3

EP 0 153 764 A2

INFORMATION PROCESSOR HAVING AN INTERRUPTION OPERATING FUNCTION

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to an information processor having an interruption operating function, and more particularly to an information processor having at least one interface circuit for indicating an interruption request from an interruption source to the information processor.

Description of the Prior Art

An interruption operating function is one of the important functions in an information processor, and is necessarily required to a microprocessor to be coupled to at least one peripheral equipment. In the prior art two types of response systems for a processing request from peripheral equipment are proposed. One is a system wherein a presence of the processing request from a peripheral equipment is monitored on a program all the time by a central processing unit (hereinafter called CPU) in an information processor. It is generally called a polling system. According to this system, CPU is not capable of working for essential data processing actively

but only monitors a pr sence of the processing requ st from the peripheral equipment, and therefore a program cannct be executed efficiently at a high speed.

The other one is an interruption system. This system is that in which CPU is not directly monitor the peripheral equipment, and only at the time when the processing request is generated from the peripheral equipment in the form of interruption to CPU, a desired data processing is carried out in the CPU according to an interruption handling program. As compared with the former system, the latter one is effective to enhance the program execution efficiency.

In this interruption mode, however, the CPU must save the processing status indicating current condition of the CPU to stack means before the interruption processing is started, in order to correctly restart the program execution which is stopped by the interruption. To save the processing status, contents of a program counter, a status word register and other registers which is holding information necessary to execute the program are to be stacked to the stack means (for example, a random access memory is generally used) through internal buses in the CPU. Further, the contents stacked to the stack means must be returned to the respective registers and the program counter through the internal buses after the interruption operation is terminated.

In the conventional information processor, these saving and returning operations (ov rh ad operations) have been necessarily performed in response to an interruption request. For example, a data transmission from a peripheral equipment to the CPU and vice versa is performed in the interruption mode. In this data transmission operation, however, the above-described program counter, the status word register and other registers are not used, nevertheless these contents must be stacked to the stack means. Therefore, a long overhead time is to be spent in the prior art information processor using the interruption system. Particularly, in the case that a common bus is employed in the CPU as the internal bus, the above-mentioned contents to be stacked can not be simultaneously transferred to the stack means, so that a very long period is required to the saving and returning.

On the other hand, an advance in semiconductor technology makes a large scale integrated circuit possible. A microprocessor which has a large number of elements on a single semiconductor chip is provided, and a plurality of peripheral equipments, such as a display unit, a printer unit, a keyboard unit, a motor, are about to be controlled by a single microprocessor. Such microprocessor requires an analog to digital or a digital to analog

cov rting function, a tim r function, a direct memory access function, a serial data transmission function, or the like. Therefore, a plurality of interface circuits for controlling communications between the CPU and peripheral equipments must be formed on a single semiconductor chip together with the CPU. In this condition, the microprocessor must receive many time of interruption request from the peripheral equipments via interface circuits. Therefore, the above-mentioned overhead time becomes longer, and the performance of the microprocessor is further reduced.

SUMMARY OF THE INVENTION

An object of the present invention is to provide an information processor which can perform an interruption operation with less overhead.

Another object of the present invention is to provide an information processor executing a data transmission between a peripheral equipment and a CPU at a high speed in an interruption mode.

Still other object of the present invention is to provide a microprocessor having a semiconductor chip on which at least one interface circuit responsive to an interruption request from an external interruption sourc and indicating a high-sp ed interruption operation to a CPU is integrated together with the CPU.

An information processor of the present invention has a central processing unit (CPU) and at l ast one interface unit. The interface unit has a receiving circuit for receiving an interruption request from an interruption source and for generating a request signal and a mode signal to the CPU. The mode signal is used to indicate an interruption mode to the CPU. There are two interruption modes in the information processor. A first mode is a normal mode in which the CPU saves information necessary to restart an execution of a program to be stopped by the interruption to stack means before the CPU executes the interruption operation and returns the saved information after the interruption operation is terminated. A second mode is a special mode (hereinafter called a macro service mode) in which the CPU executes the interruption operation without the saving of the information which is to be saved in the stack means according to the first mode.

According to the information processor of the present invention, the interruption operation is performed according to the mode signal. Particularly, when the interruption operation is performed in the second mode, the saving of contents of a program counter, a status word register, and the other registers is not required. Therefore, the interruption operation can be firstly

executed with less overhead. Furth r, the r turning of the contents to the program counter and the status word register is not required after the interruption operation is terminated. An interruption operation which does not require the stack operation is performed in the second mode. A data transmission between the CPU and a peripheral equipment can be performed in the second mode. For example, when the information processor includes an A/D converter or a D/A converter, a converted digital data or an analog data to be converted is transferred between the A/D converter or the D/A converter acted as an interface unit and the CPU in the second mode. When the information processor includes a serial communication interface unit to be coupled to a serial data manipulating peripheral equipment, a data transmission between the serial communication interface unit and the CPU in the second mode. On this case, the data transmission in th second mode is performed in a parallel form through a data bus coupled the CPU to the serial communication interface unit. When the information processor includes a realtime output port unit as the interface unit, the CPU sets a data to be outputted from the realtime output port ther in in the second mode. Further when the information processor includes an event counter unit as the interface unit, the CPU receives a content of the event counter unit which is

counted up or down whenever an event is occur d in a peripheral quipment in the second mode in response to interruption requests generated at a predetermined intervals from a timer provided in the information processor. It will be easily understood that the present invention can be also applied to another interrupt operations in which the stack operation for the program counter and the status word register is not required. Further, in the present invention an arithmetic logic unit (ALU) and an RAM in the CPU can be used in the second mode.

In the case that an information processor is realized a single chip microprocessor, an interruption request from outside of the chip and an interruption request generated in the chip can be seized as the same interruption level (hereinafter referred to as I/O requests). The I/O requests are processed by either one of two modes according to the mode signal. The first mode is a conventional interruption handling mode, processing the I/O requests according to the program manipulation. In this mode, contents of program counter, register, flag and others are saved for the interruption handling. That is, the overhead processing is executed. The second mode is the macro service by the which data transmitting between a special function register

(hereinafter called SFR) in th int rfac unit provided on the microprocessor chip and a memory (h reinafter called RAM) provided in CPU can be executed without performing the overhead processing. When the macro service comes to end, CPU restarts execution of the suspended program. Since the overhead processing is not necessary in this case, the suspended program can be performed immediately. Accordingly, the suspension of the program is not so observed from a software and looks as if a transfer instruction were inserted automatically in the program manipulation to processing.

The macro service (the second mode) is effective to minimize the number of interruption occurrence for which a software processing is influential and thus to lighten loads of the software processing. Accordingly, the macro service is employed for a simple data transmission between buffer areas of the peripheral interface unit and the memory, which has been processed by an interruption handling in the prior art. On the other hand, editing and smoothing of a series of data obtained through the macro service according to the I/O requests are processed by interruption handling of the first mode as ever before. Therefore, the microprocessor of the present invention has further means for exchanging the second mode to the first mode. This means is effective

for an interruption operation including data transmission and data handling according to a program.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram representing one embodiment of this invention;

Fig. 2 is a block diagram representing the I/O request control section 1 in Fig. 1;

Fig. 3 is a block diagram representing the execution section 2 in detail and the memory 3 in Fig. 1;

Fig. 4 is a logical drawing of the I/O request generation circuits 7, 8, 9 and the priority control circuit 10;

Fig. 5 is a block diagram of a conventional A/D converter;

Fig. 6 is a block diagram representing schematically an example of microcomputer incorporating an A/D converter therein;

Fig. 7 is a flowchart of conversion processing in Fig. 6;

Fig. 8 is a block diagram representing in detail a microcomputer having an macro service data transmission function given in an embodiment of this invention; and

Figs. 9 to 16 are block diagrams of another embodiments of the present invention, respectively.

DETAILED DESCRIPTION OF THE EMBODIMENTS

Fig. 1 is a block diagram of a main part. The embodiment comprises an I/O request control section 1 for controlling an interruption request from peripheral equipment and others, an execution section 2 for carrying out a processing according to a program and the interruption request, and a memory section 3, an interruption request signal is applied to the execution section 2, through a line 4. A mode signal is applied also to the execution section 2 for designating a mode of the interruption through a line 5. The execution section 2 and the memory section 3 are coupled to each other by an internal bus 6. When I/O requests from the peripheral equipment and others are generated and the I/O request line 4 comes to an active level, the execution section 2 suspends a program which is now being performed and accepts the I/O requests. The execution section (CPU) 2 checks a level of the I/O mode designating line 5. In the example, if the I/O designating line 5 is on a low level, the I/O request is received as a normal interruption request (the first mode), while if it is on a high level, then the request is received as a macro service.

Next, Fig. 2 is a block diagram of the I/O request control section (an interface unit) 1 in Fig. 1, and

an op ration will be described with reference thereto. Here, three I/O request sources A, B, C are taken up for description. The I/O request control section 1 has I/O request generation circuits 7, 8, 9 for generating I/O requests according to the I/O request sources A, B, C, respectively, I/O request lines 11A, 11B and 11C transfer I/O requests. I/O mode designating lines 12A, 12B and 12C transfers mode signals. Further, the control section 1 has a circuit 10 for detecting a priority level for the received I/O request signals and for indicating an interruption to the CPU 2. Thus, the I/O source with the highest priority level is selected and the I/O request signal corresponding to the selected source and the mode signal is transferred to the CPU 2 through the lines 4 and 5, respectively. Here, if the interruption operation is performed in the first mode, the I/O request source turns I/O mode designating line 12 to a low level. On the other hand, if the interruption is performed in the second mode, then it turns the designating line 12 to a high level. It is assumed that the priority level of I/O request lines 11A, 11B, 11C are given in precedence 11A > 11B > 11C.

Next, Fig. 3 represents the CPU 2 and the memory 3 in Fig. 1 in detail, and an operation will be described with reference ther to. The I/O request line 4 and

the I/O mode designating lin 5 of the I/O request control section 1 in Fig. 1 are coupled to an I/O r ceiving circuit 15. Interruption programs according to I/O requests are loaded in a program memory (ROM) 13 wherein microprograms are preliminarily stored, and the access is effected on a program counter (PC) 14. A general operation status of the CPU 2 is set in a program status word register (PSW) 16. The CPU further comprises an arithmetic logical operation unit (ALU) 17 having an arithmetic logical operation function, an instruction register 18 holding an instruction word read out of the ROM 13, an instruction decoder 19 for generating various control signals according to contents of the instruction register, and an execution control unit 20 for controlling operations of the CPU according to an output of the instruction decoder 19. The CPU has memory points (hereinafter called MP) each capable of pointing address of the RAM 3, special function register points (hereinafter called SFRP) capable of pointing SFR address in an interface unit (not shown), terminal counters (hereinafter called TC) for storing number of times of the macro service. Three register groups 21, 22, 23 each of which has MP, SFRP and TC, are prepared to macro services. Contents of the MP, SFRP and TC can be set therein according to a software.

Normally in th CPU, an instruction stor d in the ROM 13 is read out according to the program counter 14 and is transferred to the instruction register 18, the instruction transferred to the instruction register 18 is decoded to the instruction decoder 19, and the execution control unit 20 generates various control signals according to the decoding result, thus executing the program. Them, at every executions of one instruction, a value of the program counter 14 is updated to access a sequentially instruction to be required to execute the program.

Here, when the I/O receiving circuit 15 detects that the I/O request line 4 becomes an active level, an execution of the programs is to be suspended, and the level of the I/O mode designating line 5 is sampled. If the level of the line 5 is low, then the I/O receiving circuit 15 recognizes that the request is a normal interruption mode (the first mode). Under such state, contents of the program counter 14 and contents of the program status word register 16 are saved at locations in the RAM 3 designated by a stack pointer (not shown), and thereafter a leading address necessary for the interruption is set in the program counter 14. Thus the normal interruption handling program is commenced. When the interruption handling program is terminated,

th contents saved to the RAM 3 are r turned to th program counter 14 and the program status word register 16, respectively. On this first mode, the overhead operation is required.

On the other hand, if the I/O mode designating line 5 is of high level, the I/O receiving circuit 15 recotnizes that the request is the macro service, and refers to the register group corresponding to the accepted I/O source in the basis of a vector (described hereinafter) which is sent from an interface unit. At this time, contents of the program counter 14 and the program status word register 16 are not saved to the RAM 3 and is held as they are. Here, the register group 21 corresponds to the I/O source A, the register group 22 corresponds to the I/O source B, and the register group 23 corresponds to the I/O source C. If the I/O source A has the highest priority and its mode signal is of high level, the reference is made to the register group 21. In this case, the execution control circuit 20 transfers contents of the address pointed by MP to an address pointed by SFRP or contents of the address pointed by SFRP to an address pointed by MF through the bus 21. This can be effected by a control similar to the normal transmission instruction execution. Accordingly, the stack operation will never be performed

in the macro service. Then, according to the macro service a value of MP is incremented by 1 in the ALU 17 and is loaded in MP, while a value of TC is decremented by 1 in the ALU 17 and is loaded in TC, thus closing a series of macro service. However, when the TC value comes to 0, the interface unit corresponding to the I/O request source A is controlled to generate the first mode interruption request signal. As the result, the first mode interruption is generated from the interface circuit of the source A just after the macro service is terminated. Thus, the interruption handling is actuatec sonsecutively to process the data transferred by the macro service on a normal interruption handling program.

Next, Fig. 4 represents a detailed logical circuit diagram of the I/O request circuits 7, 8, 9 and the priority control circuit 10. A control circuit 100 acts as an interface unit and includes the I/O request circuit 7 and a portion of the priority control circuit 10 in Fig. 2, and control circuits 200 and 300 act as another interface circuits, and include the I/O request circuits 8 and 9 and the other part of the priority control circuit 10, respectively. Since these three control (interface) circuits operate exactly in the same way, the control circuit 100 will be taken up here

for description. The control circuit 100 comprises an I/O request flip-flop (hereinafter called I/O RQF/F) 104 which is a set/reset flip-flop (hereinafter called RSF/F) for receiving the interruption signal from the source A, an interruption mask circuit 105 for masking the received interruption signal, a condition flag 112 indicating an interruption condition, a mode F/F 113 which is RSF/F for indicating the macro service, a register 117 for storing information to the used in the macro service, a register 120 for storing a vector to be used in the normal interruption (the first mode). An I/O mode designating line 34, information line 31, an acknowledgement line 32, a reset line 33, a macro service inhibit line 35 and a read control line 36 are coupled to each control circuits to the CPU. A request line 101 from the I/O request source is compled to I/O RQF/F 104, and when the request line 101 becomes a high level, I/O RQF/F 104 is set through an OR gate 103. The interruption mask flag 105 and the mode F/F 113 are ready for setting and resetting by programs, the interruption mask flag 105 is set when the interruption request is forbidden, and the mode F/F 113 is set in response to the mode signal 12 in Fig. 2.

Here, a description will be given of an operation when the request line 101 from the I/O request source

b comes a high level, and I/O RQF/F 104 is set accordingly under th condition wh rein the interruption mask flag 105 is reset and the mode F/F 113 is reset. Since an output of the mode F/F 113 is of low level, a control input 111 for controlling the priority is of low level, and an output of I/O RQF/F 104 and an output of the interruption mask flag 105 are both of low level, an output of an AND gate 107 becomes a high level, and outputs of OR gates 108 and 110 become high level. An output of the OR gate 110 is coupled to an OR gate 210, and an output of the OR gate 210 is coupled to an OR gate 310, therefore an I/O processing request line 30 becomes a high level according to an output status of the OR gate 108. When the situation that the I/O processing request line 30 has become a high level is detected, the CPU 2 stops the program which is now executed. This line 30 is used to indicate reception of the I/O request to the CPU. The acknowledge signal (hereinafter called I/O ACK signal) is transferred from the CPU through the line 32. In response to this ACK signal, the condition F/F (hereinafter called CF/F) 112 is set through an AND gate 109. If the I/O request is generated in the other control circuits 200 and 300 in this case, since outputs of the OR gates 110 and 210 are of high level, outputs of OR gates in the control

circuits 200 and 300 corr sponding to th OR gate 108 in the control circuit 100 will never b come high l vel, and thus CF/F in the control circuits 200 and 300 is never set. That is, CF/F is a flip-flop whereby the I/O request is accepted when the I/O request source higher in priority than the I/O request of the control circuit of its own has not generated the I/O request. When CF/F 112 is set, a leading edge is detected by a leading edge detection circuit 119 to which an output of CF/F 112 is coupled, I/O RQF/F 104 is reset to provide for the next I/O request acceptance. Then, a tri-state buffer 115 is made active, and content of the mode F/F 113 is outputted to the mode designating line 34. In this case, since a low level on the mode designating line 34 is detected, the CPU recognizes the first mode, that is the normal interruption, and thus commences the stack operation for the program counter and the program status word register. After commencing the stack operation, the CPU transfers the vector read signal to the line 36 for finding a leading address of the interruption program to be processed. Since the CF/F is of a high level and the mode F/F is of a low level, an output of inverter gate 122 becomes a high level, therefore an output of AND gate 121 becomes a high level, and the tri-state buffer

119 becomes active. Accordingly, a vector information of the register 120 is outputted to a bus 31 and is sent to the CPU. In this case an output of AND gate 118 does not become a high level for the mode F/F being of a low level, and a tri-state buffer 116 does not become active. The vector data on the bus 31 is received to the CPU and a branch operation to jump the interruption program is performed. Further, a signal (hereinafter called EOP signal) 33 indicating that the I/O request processing is over is turned to a high level to reset CF/F 112. When the interruption handling program is terminated, contents saved to the RAM 3 is returned to the program counter, program status word resister and others, thus recommencing the suspended program execution.

Described next is an operation to cope with the situation in which the I/O request is generated and the I/O RQF/F 104 is set when the mode F/F 113 is set. In this case, an output of I/O RQF/F 104 and an output of the mode F/F 113 are of high level, and the control input 111 for priority is of low level, therefore an output of AND gate 106 becomes a high level, and outputs of the OR gates 108 and 110 become high level. The output of the OR gate 110 is coupled to an input of the OR gate 210, and an output of the OR gate 210 is coupled to an input of the OR gate 310, therefore

the I/O request line 30 becomes a high level unconditionally. When the situation that the I/O request line 30 has become a high level is detected, the CPU suspends the program in execution, outputs the I/O ACK signal 32, sets CF/F 112 through the AND gate 108, and resets I/O RQF/F 104 on an output of the leading edge detector 119 to which an output of CF/F 112 is coupled. Then, the output of CF/F 112 makes the tri-state buffer 115 active and outputs a content of the mode F/F 113 to the mode designating line 34. Since a high level of the I/O designating line 34 is sampled, the CPU recognizes that a macro service will be effected. Then, for necessary information to obtain for the macro service, the CPU transfers the vector read signal 36 to a high level, the AND gate 118 to a high level and the tri-state buffer 116 active, and outputs information of the register 117 holding information necessary for the macro service to the bus 31. The information outputted to the bus 31 in this case is constituted of register group selection information indicating which register group to select and information designating the direction of automatic data transmission, namely the transmission from SFR to RAM or from RAM to SFR. The CPU refers to one of the register groups 21 to 23 according to information

obtained through the bus 31. Thus, a data transmission between the CPU and a peripheral equipment corr sponding to the inhibit 100 is performed by using the bus 21. In the macro service, at every data transmission, addition and subtraction are carriod out by 1 each on MP and TC, respectively, and where the subtraction results in TC = 0, the CPU inhibits the macro service and turns the signal 35 to a high level, and resets the mode F/F 113 forcedly on a hardware. Further, since an output of CF/F 112 is of high level, I/O RQF/F 104 is set through the AND gate 102 and the OR gate 103 by turning the, inhibit signal 35 to a high level. The CPU then turns EOP signal 33 to a high level and resets CF/F 112, thus closing a series of macro service. However, since the mode F/F 113 is reset and I/O RQF/F 104 is set here, a normal interruption request (the first mode) is generated, an interruption for the transferred data is actuated consequently. Therefore, the normal interruption program is requentially executed according to data transferred through the macro service.

The above-described embodiment will be described next according to a practical application. If an A/D converter is taken up for the interface unit, then the I/O request will be generated when A/D conversion operation comes to an end. Some sampling values averaged

through arithmetic operation will be employed normally for A/D conversion value. Accordingly, the number of data to be averaged on TC is set, an address in which a result of the A/D conversion value is loaded is set on SFRP, a proper buffer area is set on the memory pointer, the mode F/F 113 is set, the interruption mask flag 105 is reset, the direction of data transmission is designated in the form of SFR to RAM to the information in the register 117, and then A/D conversion is commenced. Thus the A/D conversion value is automatically loaded in the buffer area at every ends of A/D conversion, and an interruption is generated when the operation wherein 1 is added on MP and 1 is subtracted on TC is carried out by the number of times specified to TC. Accordingly, in the interruption handling program, the number of times for interruption generation will be largely decreased simply by averaging of data loaded in the buffer area and resetting of MP and TC, thus enhancing an execution efficiency of programs.

Further, in this embodiment, an operation of the CPU is not completely stopped for control of the macro service, but the CPU itself operates for the transmission, therefore the hardware for control of the instruction execution at the time of normal program manipulation will be utilized effectively without requiring any special

hardware for macro servic . Then in the mbodiment, an example fixed in precedence is given for description, however, a processing request control variable in precedence can easily be satisfied. A data transmission only is taken up for the processing, however, an information processor exceedingly rich in flexibility will be obtainable for high application through combining a control of normal compare instruction and the like therewith without adding any special hardware further thereto.

The present invention is effective for various interface units. An example for applying to an A/D converter interface unit will be described.

An example of analog data processing on a conventional A/D converter in such a background will be described with reference to a block diagram of Fig. 5.

An A/D converter 50 comprises analog inputs 51-1 to 51-8, a multiplexer 52, an A/D channel mode register 53 having a channel designating bit 53-1 and an A/D conversion complete flag 53-2, a sample hold circuit 54, a ladder resistance network 55, a voltage comparator 56, a successive approximation register (hereinafter called SAR) 57, a controller 58, a conversion result register (hereinafter called CRR) 59, and operates for obtaining, for example, an 8-bit A/D

conversion value through a succ ssive approximation process.

First, CPU writes a value in the channel designating bit 53-1 of the A/D channel mode register 53. The A/D conversion complete flag 53-2 is then reset simultaneously. The analog input 51-1 selected through designation of the A/D channel mode register 53 is sampled in the sample hold circuit 54, and its output works as one input of the voltage comparator 56. The voltage comparator 56 sets a corresponding digital data in SAR 59 from an upper bit according to a difference between the analog input and a voltage tap selected successively by a tap decoder of the ladder resistance network 55. That is, one bit of the digital data is set through comparison carried out one time on the voltage comparator 56, and the 8-bit A/D conversion value is set in SAR 59 to the one analog input 51-1 by repeating the comparison operation. The result is latch-inputted to CRR 59, and the A/D conversion complete flag 53-2 of the A/D channel mode register 53 is set simultaneously. In response to an A/D conversion command, CPU checks the A/D conversion complete flag 53-2 at all times during operation of the A/D converter 50 for data conversion, and must wait for the A/D conversion complete flag being set. At the point in time whereat

the A/D conversion complet flag is set securely, CPU reads the A/D conversion value held in CRR 59 and operates for loading it in the memory.

Thus the microcomputer incorporating the conventional A/D converter therein must monitor an end of conversion at all times during A/D conversion, a main processing CPU must execute essentially cannot be performed all the time through, and thus an execution efficiency is not ensured satisfactorily.

To improve the above defect, there is a method wherein an interruption request is generated to CPU whenever a conversion value is loaded in CRR 59 without using the conversion complete flag 53-2. The method will be described with reference to a block diagram of Fig. 6 and a flowchart of Fig. 7. A microcomputer 60 comprises an execution division 61, a program memory 62, a data memory 63, an A/D converter 64 and an A/D interruption control division 66, which are coupled mutually through an internal bus 65. The execution division 61 comprises a program counter (hereinafter called PC) 61-1, a program status word (hereinafter called PSW) 61-2, a general purpose register set 61-3, executes a program according to an instruction code read out of the program memory 62, and loads processing data in the data memory 63. The A/D converter 64 has a CRR 64-2 for one byte, processes

an analog input 64-1 to A/D conversion, and upon loading it in CRR 64-2, makes a conversion complete signal 610 active, and then advises the A/D interruption control division 66 of the conversion having been completed. The A/D interruption control division 66 then makes an interruption request line 611 active, advises the execution division 601 of an A/D interruption request having been generated, and thus the execution division 601 executes an A/D interruption handling.

Now, a software processing procedure according to A/D interruption will be described with reference to the flowchart of Fig. 7. In the software processing, a conversion value loading area 63-1 is set in the data memory 63, and an address information for designating the conversion value loading area 63-1 and a conversion frequency information are set beforehand in conversion parameter shunt area 63-2 in the data memory 63.

When advised of a conversion having been completed, the execution division 61 executes a conversion interruption handling. First, for saving the then processing contents, each content of PC 61-1, PSW 61-2 and the general purpose register set 61-3 is set aside to a separate area prepared temporarily as a stack area 63-3. Next, an address of the conversion value loading area 63-1 is read out of the conversion parameter shunt

area 63-2 and load d in a pointer. A conversion value is then read out of CRR 64-2 and loaded in the conversion value loading area 63-1 designated by the pointer. When the conversion reaches a predetermined number of times, a series of data conversion is regarded as having been completed and so branched to a conversion procedure complete processing A. When the conversion has not reached the predetermined number of times, the address information of the conversion value loading area 63-1 and the conversion frequency data area updated and returned to the conversion parameter shunt area 63-2. Then, each content of PC 61-1, PSW 61-2 and the general purpose register set 61-3 set aside to the stack area 63-3 is returned to the original position, thus completing the conversion interruption service.

As described above, in the software processing according to interruption on the conventional micro-computer, many processings (overhead) such as interruption handling on CPU, stack operations for PC, PSW and general purpose register set to a stack register (a part of RAM), return operation from the stack register, and a return operation to a main routine are required other than transmission of the A/D conversion value. The overhead then covers a considerable extent of an overall interruption service routine, and hence the method is

also defective more or less to decrease an execution tim which can be assigned to the main processing and thus to deteriorate a throughput of CPU.

There may be conceivable, on the other hand, such buffering technique that a register equivalent to CRR 59 is incorporated plurally in parallel, and an interruption request is generated at the point in time when the A/D conversion value is loaded in all the CRR's. According to the technique, a frequency of interruption generation is decreased to $\frac{1}{n}$ by increasing the number of CRR's to n, therefore a ratio of the overhead according to interruption is decreased, and thus an execution efficiency to CPU can be enhanced for certain, however, an increase in the number of CRR's to be provided as buffers is to lead to a defect that a chip area of the microcomputer is increased inevitably.

In an information processor provided with a conversion unit (an interface unit) for generating a conversion data and a conversion complete signal according to a closing of A/D conversion, a CPU for performing a normal program manipulation and a data transmission according to the conversion complete signal, and a memory for storing a processing data and a program, the embodiment is characterized in that the CPU is

provided with an A/D conversion data processing control means for realizing an execution of an A/D conv rsion data transmission as holding a state on the way to the normal program processing, and when the A/D conversion unit generates the conversion complete signal, the CPU suspends the normal program manipulation under control of the A/D conversion data processing control means and executes the A/D conversion data transmission.

According to this invention, a processing according to an I/O request of A/D conversion completion, namely a data transmission between CRR and the data memory can be executed without interposition of a program manipulation (software). The data transmission (hereinafter called automatic data transmission) is that in which CPU suspends a program manipulation in execution when an I/O request according to A/D conversion completion is generated, and CPU itself performs the macro service.

Fig. 8 shows an embodiment of the microprocessor in Fig. 1 which includes an A/D conversion interface unit. An A/D converter 80 has CRR 80-2 and outputs an A/D conversion complete signal 80-3 to an I/O request control circuit 81. An A/D conversion interface unit 800 has the I/O request control circuit 81, an I/O request line 4 and an I/O mode designating line 5

coming ther from, and the I/O request receiving circuit 15 for controlling an operation of CPU 810. On the other hand, the memory comprises a program memory 13 for interruption handling program and others and a data memory (RAM) 3 having an A/D conversion data loading area 3-1. Further, CPU 810 comprises PC 14 for pointing an address of the instruction to execute next, an arithmetic logical operation unit (hereinafter called ALU) 17, PSW 16 for indicating a general operating state of CPU, a general purpose register set 85 for holding a data in processing, an instruction register 18 for holding the instruction to execute next, an instruction decoder 19 for interpreting contents of the instruction register 18 and generating various control signal, and an execution control unit 20 for controlling a general operation of CPU according to an output of the instruction decoder 19. Then, the register 21 necessary for the macro service is provided. The register 21 comprises a memory pointer (hereinafter called MP) 21-1 for designating an address of the data memory, an SFR pointer (hereinafter called SFRP) 21-2 for designating an address of the special_register, a terminal counter (hereinafter called TC) 21-3 for holding a transmission frequency of the A/D conversion value automatic data transmission. All the above function elements are coupled to an internal bus 210.

An operation of the macro s rvice in A/D conversion data processing will be described. When the analog input 80-1 is sampled on the A/D converter 80, and the sampled A/D conversion value is loaded in CRR 80-2, the A/D conversion complete signal 80-3 is outputted to the I/O request control division 81. The I/O request control division 81 then sends a signal through the I/O request line 4 and the I/O mode designating line 5 which are coupled to the I/O request receiving circuit 15. When the I/O request line 4 is activated, and the mode line 5 becomes a low level, then the I/O request is recognized as the first mode. While, when the mode line 5 is on a high level, the circuit 15 recognizes the I/O request as the macro service (the second mode) and sets an information in the instruction register 18. The execution control unit 2 forbids updating of PC 14 and commences the following processing without saving contents of PC 14, PSW 16 and general purpose register set 85. First, the I/O request control circuit 81 outputs information corresponding to an address of the register 21 onto the internal bus 210, and the execution control unit 20 reads the address of the register group 21. Next, the execution control unit 20 reads an A/D conversion value out of CRR 80-2 pointed by SFRP 21-2 of the register 21, loads the data in

the A/D conversion value loading area 3-1 of th data memory 3 pointed by MP 21-1, subtracts a value of TC 21-3 by 1 and loads it again in TC 21-3. A series of the above processing is to terminate the macro service one time in the A/D conversion value transmission. A series of data transmission is regarded as completed when a value of TC 21-3 becomes 0 through subtraction, and after the then A/D conversion value is loaded in the A/D conversion value loading area 3-1, the I/O request control circuit 8, makes the I/O request line 4 active again, turns the I/O mode designating line 5 to a low level to generate an A/D conversion interruption request, thus activating the A/D conversion interruption program.

As described above, the invention comprises two kinds of modes, normal software interruption (first mode) and the macro service (second mode), to cope with a conversion complete signal generated according to a closing of A/D conversion, which can be selected properly. Particularly at the time of the macro service mode, the time required for saving of PC 14, PSW 16 and general purpose register 85 which are necessary for conventional interruption handling, branch processing to the interruption program, return to the register, read of instructions, decoding and

the like can sharply be sav d, and a det rioration in throughput of CPU can be suppr ss d as low as possible.

Another embodiment of the present invention is shown in Fig. 9 in which a counter function to measure a period and width of a pulse generated from a peripheral equipment, for example, of the rotational frequency and speed of a motor, which is integrated as one function in a microcomputer chip. For measurement of period and width of a pulse, a counter comprising a counter (hereinafter called free running counter (FRC)) for counting a predetermined counting clock and a capture register (hereinafter called CPTR) for holding a counted data temporarily is used generally in a microcomputer. When a pulse signal is sent from a measuring object, the counter in such configuration loads a value indicating a counting state of the then FRC in CPTR (the operation being called capturing hereinafter), and continues a counting operation of FRC until the next pulse signal is inputted. When the next pulse signal is inputted, it captures the then value of FRC. CPTR then holds a data captured ever before. To obtain width and period of the pulse, the previous data loaded in CPTR is set aside to other register temporarily, and a difference from the data captured before will be obtained when the next FRC value is captured.

Then, for the control of high precision, a measure over nois s or an optimum control must be taken into consideration. Accordingly, there is employed a technique wherein a presence of capture generation due to noises and a degree of acceleration or deceleration are obtained through comparing information of width and period of a pulse obtained from the data captured previously with information obtained newly, or an average is obtained from data captured more than one time, thereby carrying out various controls.

In Fig. 9, a counter 95 has FRC 95-1 and CPTR 95-2, and when a value of FRC 95-1 is captured on the capture command signal 95-3, the capture end signal 95-4 is outputted. A count data processing control unit 900 has an I/O request control circuit 905, an I/O processing request line 4, an I/O mode designating line 5, and an I/O request receiving circuit 15 for controlling an operation of CPU 910. Then, the data memory 3 has the arithmetic operation result loading area 30-2, the stack area 30-3 to which the status is saved, and CPU 910 comprises PC 14 for pointing an address of the program memory 13, PSW 16 for indicating a general operating state of CPU, the general purpose rerister set 91 for holding data in processing, the ALU 17, the instruction register 18 for holding the instruction to

execute next, an instruction decoder 19 interpreting a content of the instruction register 18 and generating various control signals, and an execution control unit 20 for controlling a general operation of CPU 910 on an output of the instruction decoder 19. Further, a macro service register group 207 necessary for the macro service is provided on a part of the data memory 3. The macro service register group 207 comprises a pointer (hereinafter called I/O P) 207-1 for designating an address of the capture register, a pointer (hereinafter called BP) 207-2 for designating an address of the data memory holding the data captured previously, a memory pointer (MP) 207-3 for designating an address of the arithmetic operation result loading area 30-2 in which a subtraction result of two capture data is loaded, and a terminal counter (TC) 207-4 for holding a capture frequency. Each part is all connected to the internal bus 210.

An operation of the macro service in a count data processing will be described. When an active signal is inputted from the capture command line 95-3, the counter 95 captures the then value of FRC 95-1 in CPTR 95-2 and outputs the capture end signal 95-4. The I/O request control division 905 then turns the I/O request processing request line 4 to the I/O

receiving circuit 15 to an active lev l and also turns the I/O mode d signating line 5 to a high level.

Upon detection of the I/O mode designating line 5 being of a high level when the I/O request line 4 becomes active, the I/O receiving circuit 15 sets a macro service code in the instruction register 18 so as to process the I/O request on macro service. The execution control unit 20 forbids updating of PC 14 and commences the following processings as holding contents of PC 14, PSW 16 and the general puspose register set 91.

First, the I/O request control unit 20 outputs an address of the macro service register group 207 onto the internal bus 210, and the execution control unit 20 reads the address of the macro service register 207. Next, the execution control unit 20 reads a capture data out of CPTR 95-2 designated by I/O P 207-1 of the macro service register 207. Then, the execution control unit 20 reads the value captured previously out of the memory of the address designated by BP 207-2 of the macro-service register 207 and substracts the value captured previously from the value of CPTR. A result obtained through the subtraction is loaded in the arithmetic operation result loading area 30-2 in the data memory 3 designated by MP 207-3, and

th read value of CPTR 95-2 is loaded in the memory of the address designated by BP 207-2. Next, TC 20-4 is decreased and loaded in TC 207-4, a value of the memory pointer MP for designating the address whereat an arithmetic operation result is loaded is increased and loaded in MP 207-3. A processing of capture data subtraction and transmission in the macro service once comes to an end through a series of the above processings, however, when the value of TC 207-4 becomes 0 as the result of subtraction, a series of the macro service is regarded as having been completed, and the I/O request control circuit 905 makes the I/O processing execution request line 4 active again and also turns the I/O processing execution mode designating line 5 to a low level so as to generate a conventional interruption request. The I/O receiving circuit 15 then activates the interruption service program and executes a software processing for averaging a series of data obtained through execution of the macro service.

As described above, in this embodiment, the capture data is automatically subtracted and transferred according to the macro service when a capture end signal is generated, thus the time spent on shunt and return of PC 14, PSW 16 and the general purpose register set 91, branch processing to an interruption program, read of

instructions, decoding and the lik which ar unavoidabl in the conventional interruption service routine requiring a software processing can sharply be saved, thereby minimizing a deterioration in throughput of CPU.

In an information processing system with a data processor (microcomputer) working as a control system therefor, there are required processings for outputting a pattern information regularly and sequentially such as pattern generation of, for example, a dot character synchronized with a dot position signal in a dot printer, generation of a motor driving pattern at every step times in a step motor driving, output of a display segment information in a dynamic display control of fluorescent display tube, light emitting diode or the like. These processings are generally executed in interruption.

Another preferred embodiment of this invention will now be described with reference to Fig. 10.

The information processor comprises an I/O request processing control circuit 100 working as an input part of the output synchronizing signal 1003 of a pattern information, the CPU 1000 comprising the program counter 12, the program status word register 16, the general purpose register set 105 and the ALU 17, the program memory 13, the data memory 3 comprising the pattern output parameter saving area 30-1 and the

output patt rn loading area 30-2, which are coupl d to the internal bus 210. The I/O request processing control circuit 1001 inputting the pattern output synchronizing signal 1003 generates an I/O request signal 4 and an I/O mode designating signal 5 in response to the signal 1003. The information processor further comprises an I/O request receiving circuit 15 for controlling an operation of the CPU 1000 upon receipt of the I/O request signal 4 and the I/O mode designating signal 5 from the circuit 1001, the instruction register 18 for storing the instruction to be executed, the instruction decoder 19 for generating various control signals according to contents of the instruction register 18, the execution control unit 20 for controlling a general operation of the execution division according to an output of the struction decoder 19, a macro service register 21 comprising a memory pointer (M) 21-1 for designating an address of the data memory 3, a port pointer (hereinafter called PORTP) 21-2 for designating a port and a terminal counter (TC) 21-3 for storing a number of times of the macro service, the output unit 1006 comprising the output register 1006-2 for outputting a pattern data and the port 1006-1, and the pattern output line 1006-3. It is preferable that the pattern output division 1006 be provided with

at least two stages of registers connected in series as described hereinlater.

An arbitrary value can be set in the register 21 according to a program. In this embodiment, an address information of the pattern output data loading area 3-10 is loaded in MP 21-1 of the macro service register 21, a designating information of the output register 1006-2 is loaded in PORTP 21-2, and a pattern output frequency is loaded in TC 21-3 each beforehand.

An instruction word stored in an address of the program memory 13 designated normally by a content of the program counter 12 is set in the instruction register 18, the instruction word transferred to the instruction register 18 is interpreted on the instruction decoder 19, and thus the execution control unit 20 controls each part to realize a main program execution. The I/O request receiving circuit 15 samples the I/O request signal 4 at every completions of the instruction execution, and repeats the above operation when it is on a low level.

When the I/O request signal 302 is of a high level, a level of the I/O mode designating signal 5 is detected. When the level of the I/O mode designating signal 303 is low, a conventional interruption handling (the first mode) is executed.

On the other hand, where the I/O mode designating

signal 303 is on a high level, the I/O request receiving circuit 15 recognizes the macro service (the second mode). The CPU inhibits updating of the program counter 12, and executes the following processings without saving contents of the program counter 12 and the program status word register 16 to the stack area of the data memory 3.

(1) The execution control unit 20 reads an output pattern data out of the output data loading area 3-10 pointed by MP 21-1 of the register 310, transfers it to the output register 1006-2 pointed by PORTP 21-2, and sends it to the output line 1006-3 through the port 1006-1.

(2) A value of MP 21-1 is added by 1 by means of ALU 17 and loaded in MP 21-1 again.

(3) A value of TC 21-3 is subtracted by 1 by means of ALU 17 and loaded in TC 211-3 again.

The macro service for outputting pattern to the port is executed according to the above processings. Here, when the value of TC 21-3 becomes 0 through subtraction, the I/O request control circuit 1001 makes the I/O request signal 4 active again, and turns the I/O mode designating signal 5 to a low level at the same time, and thus generates an interruption request according to the first mode to perform a pattern output complete processing.

Fig. 11 is a block diagram of a second embodiment wherein the output register 1006-2 of the mbodim nt in Fig. 10 is made to work as a first output register, a second output register 1006-4 is provided longitudinally thereof, and thus the register is constituted in two stages.

An operation of the information processor is same as in the case of the embodiment of Fig. 10.

Generally the I/O request comes in a plurality to a processing according to priority other, and that with a low priority is subject to a control of pending acceptance. In case the I/O request with lower priority level is generated during processing of some I/O request, the low priority processing is held, but if the held I/O request is a pattern information output request, then there may arise a delay from generation of the request to pattern output.

In the processor of Fig. 11, synchronously with the pattern information output synchronizing signal 1000 being of an active level, the data held in the first output register 1006-2 is transferred to the second output register 1006-4 and outputted through the port 1006-1. Accordingly, from setting a pattern information to output to the second output register 1006-4, the pattern information can be outputted despite the I/O request being held.

In the embodiment of Fig. 11, when the I/O request with a higher priority order is completed and thus the pattern information is ready for output processing, the pattern information to output next is loaded in the second register 1006-4 through macro service operation as in the case of the embodiment in Fig. 10.

As described above, in the processor with a precedence control, an output pattern low in priority is ready for transmission even in execution of the processing high in priority from constituting the output port in two stages, thus suppressing a time lag from generation of the pattern output request to data output.

Fig. 12 shows an embodiment of the present invention having a serial data communication function.

First, referring to the block diagram of Fig. 12, an operation for I/O requests of transmission and reception in the serial data transmission is described. The serial data transmitter 1204 has the receiving buffer register 1207 (hereinafter called receiving buffer) and the transmitting buffer register 1209 (hereinafter called transmitting buffer), and outputs the reception end signal 1210 and the transmission end signal 1212 to an I/O request control unit 1200 as interruption. The I/O request control unit 1200

comprises a transmission I/O r quest circuit 2401 for generating an I/O requ-st for completion in transmission (hereinafter called transmission I/O request) upon receipt of the transmission end signal, a reception I/O request circuit 2402 for generating an I/O request for completion in reception (hereinafter called reception I/O request) upon receipt of the reception end signal, transmission/reception I/O request lines 2403-1, 2403-2, transmission/reception I/O processing mode designating lines 2404-1, 2404-2 for designating transmission/reception I/O processing modes, the I/O request line 4, the I/O mode designating line 5, transmission/reception I/O request buses 2405-1, 2405-2 and a priority control circuit 2408. When the transmitting buffer 1208 becomes empty for transmission, the transmission I/O request lines 2403-1 gets active, and if the transmission I/O request circuit 2401 is in a position to process the transmission I/O request through interruption in this case, the transmission I/O mode designating line 2404-1 is turned to a low level, and an interruption branch address is put out to the transmission I/O request bus 2405-1. Then, when the request is to be processed as the macro service, the transmission I/O mode designating line 2404-1 is turned to a high level, and an address of a first macro service register group 21 (Fig. 13)

which will be described later is outputted to th circuit 2408 through the transmission I/O request bus 2405-1. Further, when the receiving data is loaded in the receiving buffer 1007, the reception I/O request becomes active, and the reception I/O request circuit 2402 operates in the same way as above other than outputting an address of a second macro service register group 22 (Fig. 13) which will be described later to the I/O request bus.

The circuit 2408 selects the highest priority I/O request, and makes the I/O request line 4 and the I/O mode designating line 5 active.

Next, Fig. 13 is a block diagram of a CPU. In Fig. 12, the CPU comprises the unit 1200, the I/O request receiving circuit 15 for controlling an operation of the CPU upon receipt of the I/O request signal and the I/O mode designating signal from the I/O request control unit 1200 of Fig. 12, the program memory 13, the data memory 3, the program counter 14, the status word register 16, the ALU 17, a general purpose register set 506 for holding the data in processing temporarily, the instruction register 18, the instruction decoder 19, and the execution control unit 20. The CPU further comprises first and second macro service register groups 21, 22 constituted of memory pointers (MP)

21-1, 22-1 for designating an address of the data memory 3, SFR pointers (SFRP) 21-2, 22-2 for designating an address of a special register, and terminal counters (TC) 21-3, 22-3 for holding number of times of the macro service. An arbitrary value can be set in the first and second register groups 21, 22 by a software.

In this embodiment, an arrangement is such that an address of the loading area of a transmitting data is loaded in MP 21-1 of the first register group 21, a transmitting buffer address is loaded in SFRP 21-2, a transmission frequency is loaded in TC 21-3, an address of the receiving data loading area is loaded in MP 22-1 of the second automatic transmission register group 22, a receiving buffer address is loaded in SFRP 22-2, a reception frequency is loaded in TC 22-3 each beforehand by a software.

Next, an operation will be described with reference to block diagrams of Fig. 12 and Fig. 13. The CPU transfers an instruction stored in the program memory 13 which corresponds to a content of the program counter 4 to the instruction register 18, the instruction decoder 19 and the execution control unit 20 then operate according to the instruction transferred to the instruction register 18, thereby realizing a normal program execution. A value of the program counter 14

is updated to an addr ss of the instruction to execute next at every executions of one instruction. The I/O request receiving circuit 15 samples the I/O request lin 406 at every completions of the instruction, and when it is on an inactive level, the normal program operation is continued.

Next, when the I/O request receiving circuit 15 detects that the I/O request line 4 is on an active level through sampling, a level of the I/O mode designating line 407 is also sampled simultaneously. If the level of the I/O mode designating line 5 is low, then the first mode is set in the CPU. On the other hand, when the I/O mode designating line 5 is on a high level, the macro service code is set in the CPU. The execution control unit 20 forbids address updating of the program counter 14. In this case, the program counter 14, the program status word register 16 and the general purpose register set 506 are not saved to the stack with the values retained in the areas, and the following processings are commenced.

(1) The I/O request control unit 1200 outputs an address of the first macro service register 21 onto the internal bus 21.

(2) The execution control unit 20 reads the address of the first macro service register group 21 to select the register group.

(3) The execution control unit 20 reads a transmitting data from the transmitting data loading area in the RAM 3 pointed by MP 21-1 and transfers it to the transmitting buffer 1209 pointed by SFRP 21-2.

(4) A value of MP 21-1 is added by 1 by means of ALU 17 and restored in MP 21-1.

(5) A value of TC 21-3 is subtracted by 1 by means of ALU 17 and restored in TC 21-3.

The macro service data transmission comes to end through the above processings. However, when a value of TC 21-3 becomes 0 through subtraction, the transmission I/O request circuit 2401 makes the transmission I/O request line 2403-1 active again and turns the transmission I/O mode designating line 2404-1 to a low level simultaneously to generate the transmission interrupt request. In this case, accordingly, when the macro service ends, the transmission interruption program is activated, as the transmission interrupt request has been generated.

An operation when a reception I/O request is generated will be described next. When the I/O mode designating line 5 is on a low level, the reception I/O request is generated. A method for activating the reception interruption handling is the same as in the case of transmission I/O request except that the I/O

request control unit 1200 outputs a reception int rrupt branch address onto the internal bus 210. When th I/O mode designating line 5 is on a high level, the I/O request receiving circuit 15 recognizes the request to be the macro service and sets a code forcedly in the instruction register 18. In this case, the execution control unit 20 inhibits address updating of the program counter 14. The CPU commences the following processings as holding contents of the program counter 14, the status word register 16, and the general purpose register set 506.

(1) The I/O request control unit 1200 outputs an address of the second macro service register group 22 to the internal bus 210.

(2) The execution control unit 20 reads the address of the second register group 22 to select the register group.

(3) The execution control unit 20 reads a receiving data from the receiving buffer 1207 pointed by SFRP 22-2 of the second register group 22 and transfers it to the receiving data loading area in the RAM 3 pointed by MP 22-1.

(4) A value of MP 22-1 is added by 1 by means of ALU 17 and restored in MP 22-1.

(5) A value of TC 22-3 is subtracted by 1 by means of alu 17 and restored in TC 22-3.

The macro service data transmission for rec ption comes to end through the above processings. How ver, when a value of TC 22-3 becomes 0 through subtraction, the reception I/O request circuit 1200 makes the reception I/O request line 2403-2 active again as in the case of transmission, and turns the reception I/O mode designating line 2404-2 to a low level to generate a reception interrupt request. In this case, accordingly, when the macro service data transmission in reception is over, the normal reception interruption program is activated.

As described above, the serial data processor according to the invention is capable of controlling a processing mode of reception interruption and transmission interruption and another processing mode of macro service data transmission according to a reception I/O request and macro service data transmission according to transmission I/O request selectively to cope with reception I/O equest and transmission I/O request. Further, when interruption is selected on software, various data processing induced by transmission/reception I/O requests can be manipulated on an interruption program as ever before.

Another type information processor which can perform a data transmission according to a character control on hardware is described in the following.

In an information processor provided with a generation division for generating a first processing request accompanying a transmission and a second processing request accompanying a reception, a memory division for storing transmitting/receiving data and programs, and CPU for executing processings according to the first and second processing requests and programs, the CPU having a data transmission means capable of processing a data transmission corresponding to the first and second processing requests as holding its own state relating to the program execution and a data detection means for processing a predetermined detection of transmitting/receiving data.

According to this embodiment, a data transmission on character control between a transmitting buffer register or a receiving buffer register and a data memory according to I/O requests of completion in transmission and reception can be executed without interposition of a program manipulation.

According to the data transmission (hereinafter called search mode automatic data transmission), when an I/O request from peripheral hardware is generated,

CPU susp nds a current operation for program x cution and processes itself a data transmission through a control on a predetermined character search without an interposition of other program manipulations as holding the then CPU status (program counter and program status word) and the data in a general purpose register set. Further, for protection of the data memory on reception side from a transmission runaway capable of arising with a consecutive processing, a maximum data transmission/reception frequency is set other than detection of a coincidence with a search character, and thus a serial transmission or reception will be completed according to a coincidence with the maximum data transmission/reception frequency regardless of a miss, if any, in the detection of a coincidence with the search character. As described, in the search mode automatic transmission, a generation frequency of the transmission complete interruption or the reception complete interruption which is processed substantially on software is minimized, CPU is enlightened for software processing, and a double data transmission control is realized through search character control and maximum transmission frequency control.

Further in the search mode automatic data transmission, a processing after completion in a

predetermined number of transmissions and receptions according to a coincidence of the transmitting/receiving data with a predetermined search character, or a coincidence of the data transmission frequency with a maximum data transmission/reception frequency is met by interruption.

Fig. 14 is a block diagram representing the function. The serial data transmitter 140 has the transmitting buffer register 1400-1 and the receiving buffer register 1400-2, and outputs the transmission complete signal 1400-3 and the reception complete signal 1400-4 to an I/O request control division 1401. The execution division comprises an I/O request acceptance division 202 for controlling an operation of the execution division upon receipt of an I/O processing execution request line 201-1 and an I/O processing execution mode designating line 201-2 from the I/O request control division 201, the program memory 103 for loading programs such as interruption handling program and the like, the data memory 104 for holding transmission/reception processing data, the program counter 102-1 for pointing an address of the program to execute next, an arithmetic logical operation unit (hereinafter called ALU) 203 having an arithmetic logical operation function, the program status word 102-2 indicating a general operating

stat of the execution division, the general purpose register set 102-3 for holding data in processing, an instruction register 204 for holding the instruction to execute next, an instruction decoder 205 for interpreting contents of the instruction register 204 and generating various control signals, and an execution control division 206 for controlling a general operation of the execution division according to an output of the instruction decoder 205. Further, the execution division has first and second search mode automatic transmission register groups 207, 208, and the search mode automatic transmission register groups 207, 208 comprise search character registers (hereinafter called SCR) 207-1, 208-1 for holding a value of the search character for detecting a serial transmission/reception completion, memory pointers (hereinafter called MP) 207-2, 208-2 for designating an address of the data memory, SFR pointers (hereinafter called SFRP) 207-3, 208-3 for designating an address of the special register, terminal counters (hereinafter called TC) 207-4, 208-4 for holding a transmission frequency of the search mode automatic data transmission, and all the above function units are connected to the internal bus 105.

With reference to the block diagram, an operation

of the search mode automatic data transmission in the serial transmission and reception of th invention will b described below.

In regard to a transmission I/O request, when the transmitting buffer register 100-1 becomes empty from sending a transmitting data externally, the serial data transmitter 100 outputs first the transmission complete signal 100-3 to the I/O request control division 201. The I/O request control division 201 then outputs a signal to the I/O request acceptance division 202 through the I/O processing execution request line 201-1 and the I/O processing execution mode designating line 201-2. Upon detection of the I/O processing execution request line 201-1 being active, the I/O request acceptance division 202 samples a level of the I/O processing execution mode designating line 201-2, and if the level is low, then the I/O request acceptance division 202 recognizes the I/O request as an interruption handling, and sets an interruption handling code in the instruction register 204. The execution control division 206 then forbids updating of an address of the program counter 102-1 and sets aside values of the program counter 102-1, the program status word 102-2 to the data memory 104. This comes in a normal interruption handling. Next,

the I/O request control division 201 outputs a branch address of the transmission interruption program onto the internal bus 105, and the execution control division 206 transfers the interruption branch address to the program counter 102-1, thereby activating the transmission interruption handling program. When the interruption service program comes to end, data set aside to the data memory 104 are returned to the program counter 102-1 and the program status word 102-2, and a program manipulation on the way to execution is recommenced.

On the other hand, if the I/O processing execution mode designating line 201-2 is on a high level, the I/O request acceptance division 202 recognizes the I/O request as a search mode automatic data transmission request and sets a search mode automatic data transmission processing code in the instruction register 204. The execution control division 206 forbids updating of an address of the program counter 102-1 and commences the following processing as holding values of the program counter 102-1, the program status word 102-2 and the general purpose register set 102-3.

First, the I/O request control division 201 outputs an address of the first search mode automatic transmission register group 207 onto the internal bus 105, and the

ex cution control division 206 reads the address of the first search mode automatic transmission register group 207 to select the register group. Next, the ex cution control division 206 reads a transmitting data out of the transmitting data loading area 104 pointed by MP 207-2 of the first search mode automatic transmission register group 207, performs a subtraction with a value of SCR 207-1 by means of ALU 203, and if the result is not 0, the transmitting data is transferred to the transmitting buffer register 100-1 pointed by SFRP 207-3. Then, a value of MP 207-2 is added by 1 by means of ALU 203 and reloaded in MP 207-2, and a value of TC 207-4 is subtracted by 1 and reloaded in TC 207-4.

A series of the above processing is to close the search mode automatic data transmission once in the serial transmission, however, when the result obtained through subtracting the value of SCR 207-1 from the transmitting data comes to 0, that is, a coincidence of the search character with the transmitting data is detected, or the value of TC 207-4 becomes 0 through subtraction, a series of data transmission is regarded as completed, the transmitting data is transferred to the transmitting buffer register 100-1, the I/O request control division 201 then makes the

I/O processing execution request line 201-1 active again, turns the I/O processing execution mode designating line 201-2 to a low level to generate a transmission interruption request, thus activating the transmission interruption program.

An operation for receiption I/O request will be described next. When the receiving buffer register 100-2 becomes full, the serial data transmitter 100 outputs the reception complete signal 100-4 to the I/O request control division 201. The I/O request control division 201 then outputs the I/O processing execution request line 201-1 and the I/O processing execution mode designating line 201-2 to the I/O request acceptance division 202. The I/O request acceptance division 202 detects that the I/O processing execution request line 201-1 is active and samples a level of the I/O processing execution mode designating line 201-2, and if it is of a low level, then the I/O request acceptance division 202 recognizes the I/O request as an interruption handling and activates the reception interruption handling program. The processing operation in this case is the same as in the case of transmission I/O request except that the I/O request control division 201 outputs a branch address of the reception interruption service program onto the internal bus 105.

When the I/O processing execution mode designating line 201-2 is of a high level, the I/O request acceptance division 202 recognizes the I/O request as a search mode automatic data transmission request, and sets a search mode automatic data transmission processing code in the instruction register 204. The execution control division 206 forbids updating of an address of the program counter 102-1, and commences the following processing as holding values of the program counter 102-1, the program status word 102-2 and the general purpose register set 102-3.

The I/O request control division 201 outputs an address of the second search mode automatic transmission register group 208 onto the internal bus 105, and the execution control division 206 reads the address of the second search mode automatic transmission register group 208 to select the register group.

Next, the execution control division 206 reads a receiving data out of the receiving buffer register 100-2 pointed by SFRP 208-3 of the second search mode automatic transmission register group 208, performs a subtraction with a value of SCR 208-1 by means of ALU 203, and if the result is not 0, the receiving data is transferred to the receiving data loading area 104 pointed by MP 208-2. Then a value

of MP 208-2 is added by 1 by means of ALU 203 and reloaded in MP 208-2, and a value of TC 208-4 is subtracted by 1 and reloaded in TC 208-4.

A series of the above processing is to close the search mode automatic data transmission once in the serial transmission, however, when the result obtained through subtracting the value of SCR 208-1 from the receiving data comes to 0, that is, a coincidence of the search character with the receiving data is detected, or the value of TC 208-4 becomes 0 through subtraction, a series of data reception is regarded as completed, the receiving data is loaded in the receiving data loading area 100-2, the I/O request control division 201 makes the I/O processing execution request line 201-1 active again as in the case of transmission, the I/O processing execution mode designating line 201-2 is then turned to a low level, and thus the reception interruption request is generated to activate the reception interruption program.

As described above, according to this invention, a serial information processor is realizable which operates on two kinds of means to cope with transmission/reception I/O requests, if any, through a normal software interruption handling and a search mode

automatic data transmission, which can be selected properly by software.

Particularly in the search mode automatic data transmission, the time required for shunt and return of CPU status and data and execution of instructions can sharply be decreased by eliminating a software processing by interruption, and further a rapid and reliable data transmission can be ensured by a double control on character and maximum transmission frequency.

Then, in a serial data transmission system according a multitude of serial information processors with an address allocated beforehand as another embodiment, a destination serial information processor will be selected without interposition of an interruption handling on software and thus a serial data transmission can be commenced, by performing a search mode automatic data transmission processing with an address information of each serial information processor set as a search character and the search character set as a start information for the serial data transmission. Accordingly, it goes without saying that a serial data transmission system configuration according to the search mode automatic data transmission processing for which the search character having an address information of each serial information processor and the search character

having an end information of the serial data transmission are both combined can easily be realized.

Still another embodiment of this invention is shown in Fig. 15 a wherein DC motor driving gear is controlled. In a DC motor driving gear provided with a timer for generating a processing request at desired time intervals, a counter for counting an input pulse, a D/A converter, a memory for storing a program and various data, CPU for executing selectively a processing according to the processing request and a processing according to the program, the invention is characterized in that when the timer generates the processing request, CPU suspends an execution of the program, creates a speed deviation data without setting aside the then status, and transmits it to the D/A converter.

Fig. 15 represents an embodiment. A timer 2030 has a timer register 2030-1 and outputs a timer I/O request signal 2030-2 to an I/O request control division 2020-2. A conversion data processing control division 2020 has the I/O request control division 2020-2, an I/O processing execution request line 2020-3, an I/O processing execution mode designating line 2020-4 and an I/O request acceptance division 2020-1 for controlling an operation of CPU 2000. CPU 2000 comprises a program counter 2000-1 for pointing an address of the program

to execute next, an arithmetic logical op ration unit (ALU) 2000-4 having an arithmetic logical operation function, a program status word register 2000-2 for indicating a general operating state of CPU, a general purpose register set 2000-3 for holding data in processing, an instruction register 2000-5 for holding the instruction to execute next, an instruction decoder 2000-6 for interpreting contents of the instruction register 2000-5 and generating various control signals, an execution control division 2000-7 for controlling a general operation of CPU on an output of the instruction decoder 2000-6. A macro-service register group 2070 necessary for the macro-service is set on a part of a data memory 2060. The macro-service register group 2070 comprises a memory pointer (MP) 2070-1 for designating an address of a command value table area 2060-1 set beforehand in the data memory 2060, a pointer (SFRP) 2070-2 for designating an address of a D/A register 2050-1, and a terminal counter (TC) 2070-3 for holding a data transmission frequency to the D/A register by the macro-service. A counter 2040 has a count register 2040-1 and a count capture register 2040-2, the count register 2040-1 counts an encoder pulse inputted to a count input line 2040-3, transmits a count register value to the count capture register

2040-2 in response to the timer I/O request signal 2030-2, and is cleared immediately thereafter. A D/A converter 2050 has the D/A register 2050-1 and outputs a voltage corresponding to a value set in the D/A register 2050-1 through a D/A converter output line 2050-2. The output is then impressed on the DC motor as an analog signal.

Operations of the first mode and the second mode (macro-service) will be described with reference to Fig. 15. When the time corresponding to the value set in the timer register 2030-1 from the timer 2030 passes, the timer I/O request signal 2030-2 becomes active. Recognizing the timer request signal 2030-2 as active, the I/O request control division 2020-2 makes the I/O request acceptance division 2020-1 drive the I/O request line 2020-3 and the I/O mode designating line 2020-4. Upon detection of the I/O request line 2020-3 being active, the I/O request acceptance division 2020-1 samples a level of the I/O mode designating line 2020-4. If it is on a low level, then the I/O request acceptance division 2020-1 recognizes the I/O request as an interruption handling request, and sets an interruption handling code in the instruction register 2000-5. The execution control division 2000-7 then forbids updating of an address of the program counter 2000-1 and sets

aside valu s of the program counter 2000-1, the program status word register 2000-2 to the data memory 2060. Next, the I/O request control division 2020-2 outputs a branch address of a timer interruption program onto an internal bus 2080, and the execution control division 2000-7 sets the interruption branch address in the program counter 2000-1, thereby activating the timer interruption program. When the interruption service program ends, data set aside to the data memory 2060 are returned to the program status word register 2000-2 and the program counter 2000-1, and the suspended program manipulation is recommenced. The above refers to the first mode.

On the other hand, if the I/O processing execution mode designating line 2020-4 is on a high level when the I/O request line 2020-3 becomes active, the I/O request acceptance division 2020-1 recognizes the I/O request as a macro-service request, and sets a macro-service code in the instruction register 2000-5. The execution control division 2000-7 commences the following processes as holding the status of the program counter 2000-1, the program status word register 2000-2 and the general purpose register set 2000-3. First, the I/O request control division 2020-2 outputs an address of the macro-service register group 2070, the execution

control division 2000-7 reads the address and selects the register group. Next, the execution control division 2000-7 reads a data of the address pointed by MP 2070-1, or the then command value out of the command value table area 2060-1 and performs a subtraction with the count capture register 2040-2 by means of ALU 2000-4. The result is transferred to the address pointed by SFRP 2070-2, namely the D/A register 2050-1. Then, a value of MP 2070-1 is added by 1 by means of ALU 2000-4 and reloaded in MP 2070-1. On the other hand, a value of TC 2070-3 is subtracted by 1 and reloaded in TC 2070-3.

The voltage to be impressed on the motor is thus changed automatically to a voltage whereby the motor rotational speed will follow the command value referred to by MP 2070-1 by updating of the D/A register according to I/O request from the timer through a series of the above processings. When TC 2070-3 becomes 0 through the subtraction, the I/O request control division 2020-2 regards a series of acceleration or deceleration processing as having been completed and makes the I/O request line 2020-3 active again. Further, the I/O mode designating line 2020-4 is turned to a loe level, a timer interruption is generated forcedly, and the software is advised of an end of a series of macro-service.

As described above, according to this embodiment, a generation of the interruption for which a software processing is required will be decreased, and the time spent on shunt and return of the program counter, program status word and general purpose register, branch processing to the interruption handling program, read of the instruction, decoding and the like can sharply be saved. A deterioration in throughput of CPU can be minimized consequently. Further, the command value need not be updated, or for example, MP 2070-1 is not added by 1 and the command value is taken from the same table at all times for a constant-speed rotation, thereby controlling it easily. As described, the DC motor driving fear according to this embodiment has an epoch-making D/A conversion data processing control means, thus ensuring a very high feasibility.

Still another embodiment which is suitable to a serial data transmission requiring a time monitoring. A serial interface generates an interruption whenever one data is transmitted or received, a serial data is transmitted or received on a software processing of CPU, and a timer allocated to the serial interface operates for a time monitoring of normal serial transmission/reception, that is, a reception completion at destination within a constant period of time in a

serial transmission and a presence of received data within a constant period of time in a serial reception are detected.

In an information processor provided with a serial data transmitter for generating a request to send for transmission and a request to receive for reception, a timer for generating a measurement complete processing request for completion of a time measurement, a memory division for storing transmission/reception data and programs, CPU for executing processings according to the request to send, request to receive and measurement complete processing request, the invention comprises a data transmitting means for transmitting a data corresponding to the request to send and request to receive as holding a status of CPU for program manipulation, a serial transmission/reception abnormity detecting means according to the measurement complete proceing request of the timer, CPU monitoring a predetermined transmission/reception data processing by the data transmitting means and the transmission/reception data processing by the serial transmission/reception abnormity detecting means when the serial data transmitter generates the request to send and request to receive.

Fig. 16 is a block diagram representing the above-mentioned information processor. In the embodiment,

the s rial data transmitter 601 has the transmission line 6061, th transmission buffer register 6011 and the reception line 6062, the reception buffer register 6012, and is further provided with CTS line 6063, DSR line 6064 as transmission/reception control signals to output TxBE 6065, RxBF 6066 and CTS notice signal 6067 to an I/O request control division 6031. The timer 606 is constituted of one set of the timer unit timer (0) 16 and the decremeter 6022. The timer (0) 6016 consists of the timer 0 modulo resister 6018 and the timer 0 timer register 6019, and outputs the timer 0 count complete signal 6069 to the I/O request control division 6031.

The I/O request control division 6031 has a service control register group 6041 consisting of an I/O channel register having an address information of peripheral hardware and a monitoring function enable bit, which comprises a transmission channel register 6042, a transmission monitoring function enable bit 6043 and a reception channel register 6044, a reception monitoring function enable bit 6045.

The execution division comprises an I/O request acceptance division 6032 for controlling operation of the execution division upon receipt oi an I/O processing execution request line 6071 and an I/O processing

execution mod designating lin 6072 from the I/O request control division 6031, the program memory 604 for loading programs such as interruption service program and the like, the data memory 605 for holding a transmission/ reception processing data, PC 6013 for pointing an address of the program to execute next, an arithmetic logical operation unit (ALU) 6033 having an arithmetic logical operation function, PSW 6014 indicating a general operating state of the execution division, the general purpose register set 6015 for holding data in processing, an instruction register 6034 for holding the instruction to execute next, an instruction decoder 6035 for interpreting contents of the instruction register 6034 and generating various control signals, and an execution control division 6036 for controlling a general operation of the execution division on an output of the instruction decoder 6035. Further, the execution division has first and second automatic transmission register groups 6037, 6038 with monitoring function required for automatic data transmission provided with a monitoring function by the timer (hereinafter called macro-service with monitoring function), and the macro-service register groups 6037, 6038 comprise respectively monitoring timer modulo divisions (hereinafter called WDM) 6046, 6051 for software monitoring timers mapped on the memory,

monitoring timer counter divisions (hereinafter called WDC) 6047, 6052 and SFR pointers (hereinafter called SFRP) 6048, 6053 for designating an address of a special register, terminal counters (TC) 6049, 6054 for holding processing frequency of the automatic transmission, memory pointers (MP) 6050, 6055 for designating an address of the data memory, and all the function units are connected to the internal bus 607.

An operation of the macro service data transmission with monitoring function in the serial transmission/reception will be described, next.

For transmission I/O request processing, first a count value is set in the timer 0 modulo register 6018 and WDM 6046 of the first macro service register group 6037 with monitoring function to obtain a predetermined monitoring time interval (time from transmission completion to destination's reception completion). The predetermined monitoring time can be expressed by the product of a count value of the timer 0 modulo register 6018 and a count value of WDM 6046. When the transmission buffer register 6011 becomes empty from having a transmission data sent externally, the serial data transmitter 1 output TxBE 6065 to the I/O request control division 6031.

The I/O request control division 6031 then outputs

the I/O request line 6071 and the I/O mode designating line 6072 to the I/O request acceptance division 6032. Upon detection of the I/O request line 6071 being active and also the I/O mode designating line 6072 being on a low level, the I/O request acceptance division 6032 recognizes the I/O request as an interruption handling and executes a normal interruption handling accompanying shunt and return of PC 6013, PSW 6014 and the general purpose register set 6015.

On the other hand, if the I/O mode designating line 6072 is of high level, the I/O request acceptance division 6032 recognizes the I/O request as macro service data transmission request with monitoring function, and sets the processing code in the instruction register 6034.

The execution control division 6036 performs the next processing as holding values of PC 6013, PSW 6014 and the general purpose register set 6015. First, the I/O request control division 6031 outputs an address of the first macro service transmission register group 6037 with monitoring function onto the internal bus 607 through the transmission channel register 6042, the execution control division 6036 then reads the address to select the first register group 6037 with monitoring function. Next, the execution control division 6036

reads a transmission data out of the transmission data loading area A pointed by MP 6050 of the first register group 6037, transfers it to the transmission buffer register 11 pointed by SFRP 6048, further transfers a value of WDM 6046 to WDC 6047, turns the transmission monitoring function enable bit 6043 in the service control register group 6041 of the I/O request control division 6031 to "1", and then sends a start signal to the timer (0) for controlling the monitoring software timer. Then, a value of MP 6050 is added by "1" through ALU 6033, reloaded in MP 6050, and a value of TC 6049 is subtracted by "1" and reloaded in TC 6049, thus closing the macro service data transmission.

Then, when the value of TC 6049 becomes "0" through the subtraction, a series of data transmission is regarded as completed, the I/O request control division 6031 makes the I/O request line 6071 active again and turns the I/O mode designating line 6072 to a low level in this case to generate a transmission interruption request, thus activating the transmission interruption service program.

The timer (0) 6016 having started according to the macro service data transmission on transmission I/O request outputs the timer 0 count complete signal 6069 to the I/O request control division 6031 according to

a value set in the timer 0 module register 6018 at very predetermined time interval. The I/O request control division 6031 turns the I/O processing execution request line 6071 active and the I/O processing execution mode designating line 6072 to a high level, and advises the I/O request acceptance division 6032 of a timer 0 automatic data transmission.

For the timer 0 macro service data transmission, a series of the following processing is executed by the I/O request control division 6031 and the execution control division 6036.

1) An address information of a first channel register (or the transmission channel register 6042) in the service control register group 6041 of the I/O request control division 6031 is output onto the internal bus 607.

2) The execution control division 6036 reads the address to select the first register group 6037 with monitoring function, and subtracts a value of WDC 6047 by 1 through ALU 6033.

3) The transmission monitoring function enable bit 6043 is checked, and if it is "1", a subtraction result is reloaded in WDC 6047, but if "0", then not reloaded. (Since the transmission monitoring function enable bit 6043 is "1" in this case, the value of WDC 6047 is decr as d by 1.)

4) The ex cution control division 6036 advises the I/O request control division 6031 of the subtraction having been over one time, and the I/O request control division 6031 outputs an address information of the next channel register onto the internal bus 607.

Only WDC of the I/O request source for which the monitoring function enable bit is "1" is decreased by 1 by repeating a series of the above processing 1) to 4). Further, in case the value of some WDC becomes 0 after a series of subtraction to all the I/O request sources is completed, the I/O request control division 6031 makes the I/O request line 6071 active again and also turns the I/O designating line 6072 to a low level this time, and activates the timer 0 interruption handling.

Where a normal serial transmission has been performed, the serial data transmitter 601 is advised of the destination's reception completion within a predetermined time by making CTS 6063 active. When CTS 6063 becomes active, the serial data transmitter 1 sends CTS notice signal 6067 to the I/O request control division 6031, the I/O request control division 6031 then makes the I/O request line 6071 to the I/O request acceptance division 6032 active and also turns the I/O

mode d signating line 6072 to a high level, thereby performing CTS automatic data transmission. With th transmission monitoring function enable bit 6043 in the service control register group 6041 of the I/O request control division 6031 kept at "0", this is to stop the subtraction of WDC 6047 according to the timer 0 macro service data transmission.

Then, in case an abnormity is caused on the serial transmission and thus CTS does not become active within a predetermined time, subtraction and stop of WDC 6047 according to CTS automatic data transmission are not carried out, and as the result of a subtraction of WDC 6047 on the timer (0) 6016 according to macro service data transmission, the value of WDC 6047 becomes "0", and the timer 0 interruption handling is activated, thereby detecting the abnormity on the serial transmission.

A reception I/O request will be described, next,

As in the case of transmission I/O request, a count value is set beforehand in the timer 0 modulo register 6018 and WDM 6052 of the second register group 6038 with monitoring function so as to obtain a predetermined monitoring time interval (data reception interval). When the reception buffer register 6012 becomes full, the serial data transmitter 1 makes DSR active to advise the source of reception completion

and outputs RxBF 6066 to th I/O request control division 6031. The I/O request control division 6031 then outputs the I/O request line 6071 and the I/O mode designating line 6072 to the I/O request acceptance division 6032.

The I/O request acceptance division 6032 detects that the I/O processing execution request line 6071 is active and samples a level of the I/O request processing execution mode designating line 6072 at the same time, and if it is on a low level, then the I/O request acceptance division 6032 recognizes the I/O request as an interruption handling, and the reception interruption service program is activated. When the I/O processing execution mode designating line 6072 is on a high level, the I/O request acceptance division 6032 recognizes the I/O request as the marco service data transmission request with monitoring function, and sets the processing code in the instruction register 6034. The execution control division 6036 commences the following processing as holding values of PC 6013, PSW 6014 and the general purpose register set 6015.

The I/O request control division 6031 outputs an address of the second register group 6038 with monitoring function onto the internal bus 607, and the execution

control division 6036 reads the address to select the second register group 6038 with monitoring function.

Next, the execution control division 6036 reads a reception data out of the reception buffer register 6012 pointed by SFRP 6053 of the second automatic transmission register group 6038 with monitoring function, transfers it to the reception data loading area D pointed by MP 6055, transfers further a value of WDM 6051 to WDC 6052, turns then the reception monitoring function enable bit 6045 in the service control register group 6041 of the I/O request control division 6031 to "1", and sends a start signal to the timer (0) 16. The value of MP 6055 is then added by "1" and the value of TC 6054 is subtracted by "1" through ALU 6033, each is returned, and thus the marco service data transmission is closed. When the value of TC 6054 then becomes "0" through subtraction, a series of the data reception is regarded as completed, and the reception interruption program is activated as in the case of transmission.

The timer (0) 6016 having started according to the serial reception macro service data transmission outputs the timer 0 count complete signal 6065 to the I/O request control division 6032 at every predetermined time intervals according to a value

set in the timer 0 modulo register 6018. The I/O request control division 6031 then makes the I/O request line 6071 active, turns the I/O mode designating line 6072 to a high level, and then advises the I/O request acceptance division 6032 of the timer 0 macro service data transmission.

The timer 0 automatic data transmission decreases WDC of the macro service data transmission register group corresponding to the I/O request source for which the monitoring function enable bit in the service control register group 6041 of the I/O request control division 6031 all by "1" as in the case of transmission.

In regard to the serial reception, the execution control division 6036 reads an address of the reception channel register 6044 to select the second register group 6038 with monitoring function, decreases a value of WDC 6052 by "1" through ALU 33, and returns the result to WDC 6052.

When the value of some WDC becomes 0 after a series of the subtraction to all I/O request sources is completed as in the case of transmission, the I/O request control division 6031 makes the I/O request line 6071 active again and turns the I/O mode designating line 6072 to a low level this time, thus activating the timer 0 interruption handling.

In case the serial reception is performed at normal time intervals, WDC 6052 is initialized whenever the data is received through automatic data transmission according to a reception I/O request. However, if there arises an abnormity on serial reception operation and thus the next data is not received within normal time interval, a result of subtraction of the timer (0) 6016 through macro service data transmission comes to "0" as WDC 6052 is not initialized, and thus a timer 0 interruption is activated, thereby detecting the abnormity of the serial reception on CPU.

WHAT IS CLAIMED IS:

1. An information processor comprising a program memory, a central processing unit performing a normal program operation and an interruption operation and at least one interface unit to be coupled to a peripheral equipment for transferring data between said central processing unit and said peripheral equipment, said central processing unit including a program counter for reading instructions out of said program memory, a status word register storing information which indicates an execution condition on said central processing unit and a stack memory for saving a content of at least one of said program counter and said status word register, said interface unit generating a first mode signal designating a normal interruption and a second mode signal designating a special interruption, said central processing unit executing the normal interruption after the content of at least one of said program counter and said status word register is saved in said stack memory in response to said first mode signal, and executing the special interruption without saving the contents of said program counter and said status word register in response to said second mode signal.

2. An information processor as claimed in Claim 1, said interface unit furth r comprising means for activating said first mode signal after said central processing unit terminates the operation of said special interruption.

3. An information processor as claimed in Claim 1, wherein said interface unit has an analog to degital conversion function.

4. An information processor as claimed in Claim 1, wherein said interface unit has a digital to analog conversion function.

5. An information processor as claimed in Claim 1, wherein said interface unit has a data communication function.

6. An information processor as claimed in Claim 5, wherein said interface unit executes at least one of a data transmission and a data reception.

7. An information processor as claimed in Claim 5, wherein said interface unit executes the data communication according to a timer output.

8. An information proc ssor as claimed in Claim 1, wherein said interface unit has a motor driving function.

9. An information processor as claimed in Claim 1, wherein said interface unit is coupled to a printer and has transfers data to be printed to the printer.

10. An information processor as claimed in Claim 1, wherein said interface unit is coupled to a display means and transfers data to be displayed to the display means.

11. An information processor as claimed in Claim 1, wherein said interface unit is coupled to a keyboard means and transfers a scanning signal to the keyboard means.

12. An information processor comprising a program execution unit executing a program operation and an interruption operation, a stack means for saving information necessary to restart a program operation which is stopped by the interruption operation, means for generating a mode signal in response to request of the interruption, and control means controlling

the interruption operation in such mann r that said program execution unit starts th interruption operation without saving the information in said stack means according to said mode signal.

13. An information processor comprising a counter for counting a clock, storage means for temperarily storing a value of the counter, means for generating an interruption request signal, a memory for storing a program, a central processing unit for selectively executing a first processing according to said interruption request signal and a second processing according to said program, said central processing unit having a counter control means for performing arithmetic operation and transmission of a data stored in said storage means as holding a status in execution of said second processing, said counter control means performing said first processing when said counter generates said interruption request signal.

14. An information processor comprising an A/D conversion processing means for generating a converted data and a conversion end signal according to a closing of A/D conversion, a memory for storing a program, a central processing unit executing a processing according

to said conversion end signal and a processing according to said program selectiv ly, an A/D conversion control means transf rring said converted data from said A/D conversion processing means to said central processing unit in an interruption mode without stack operation for saving information necessary to restart said program which is stopped by the interruption mode.

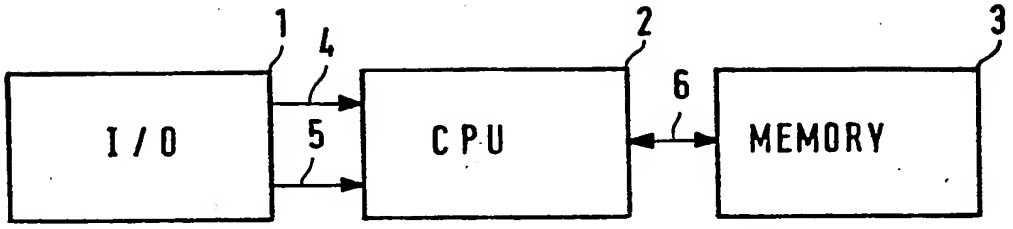


FIG. 1

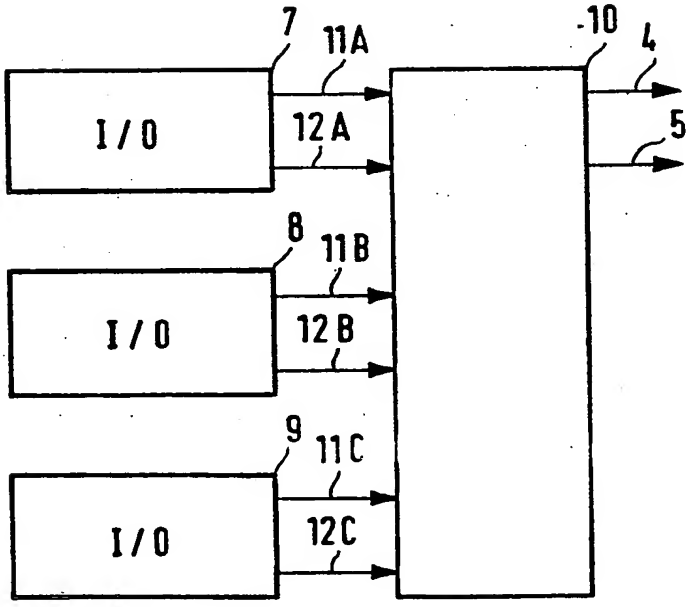


FIG. 2

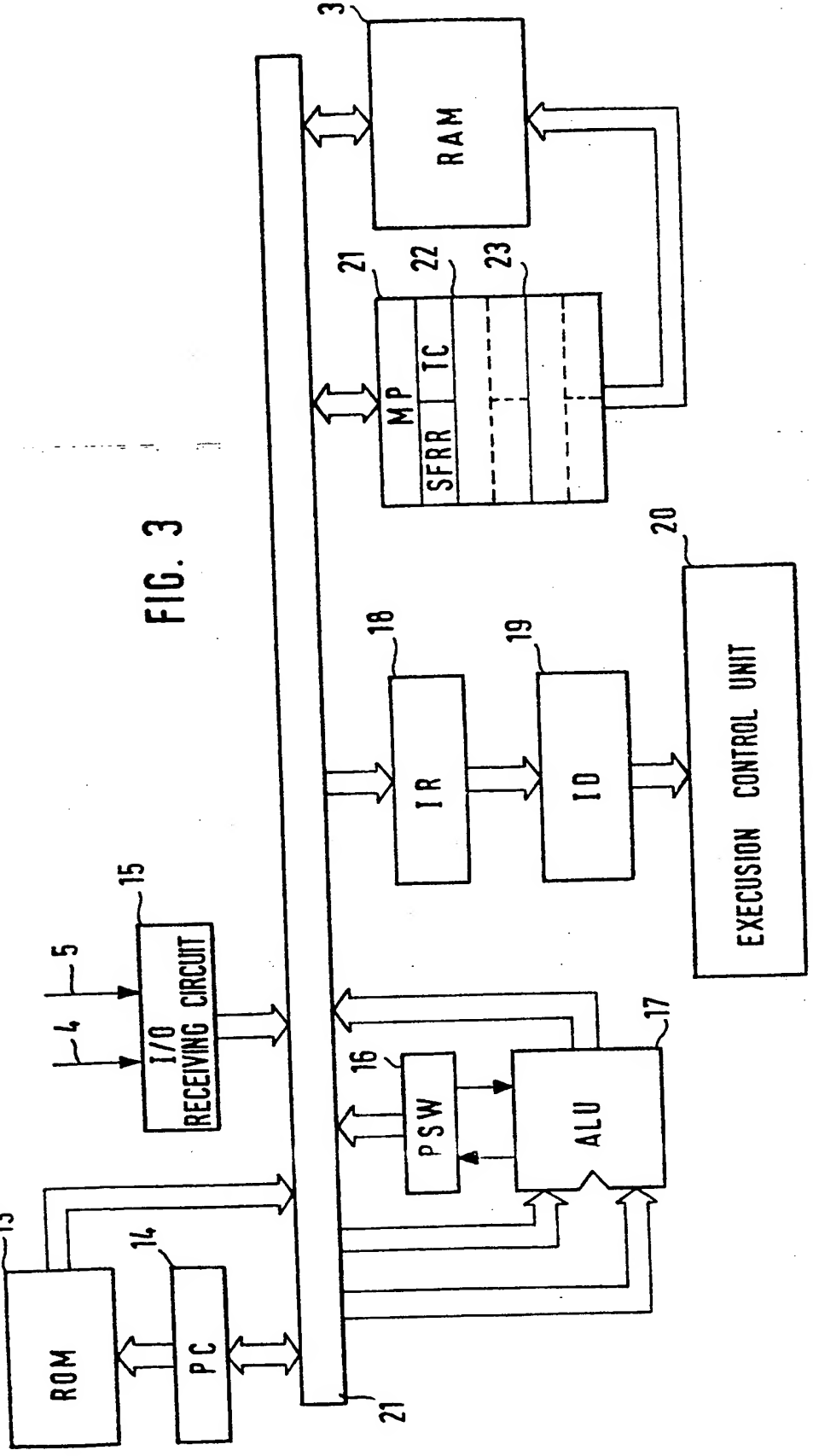

3
RAM
21
22
23
MP
TC
SFRR
20
EXECUSION CONTROL UNIT
18
IR
19
ID
15
5
4
I/O
RECEIVING CIRCUIT
16
PSW
17
ALU
13
ROM
14
PC
21

FIG. 3

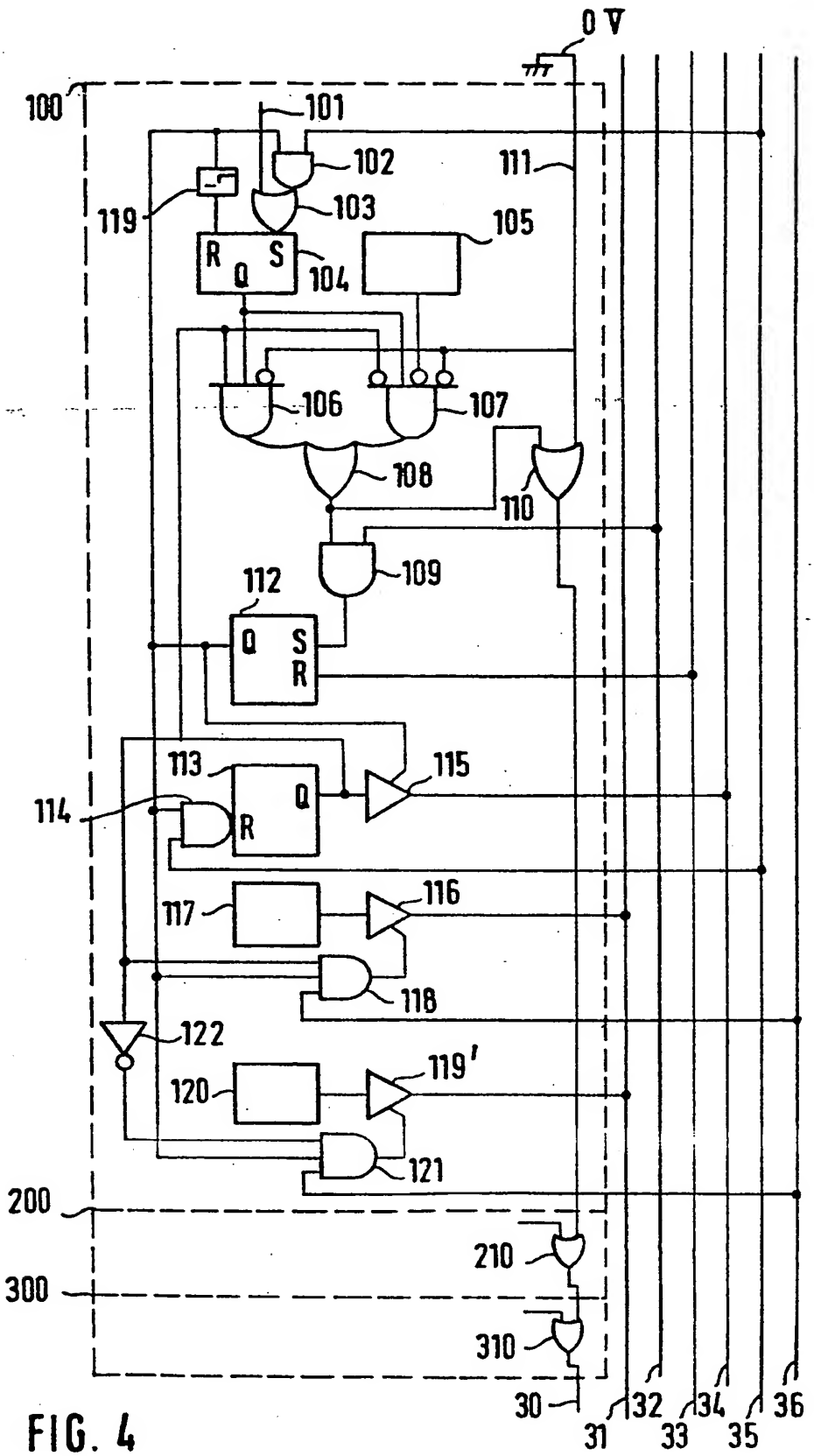
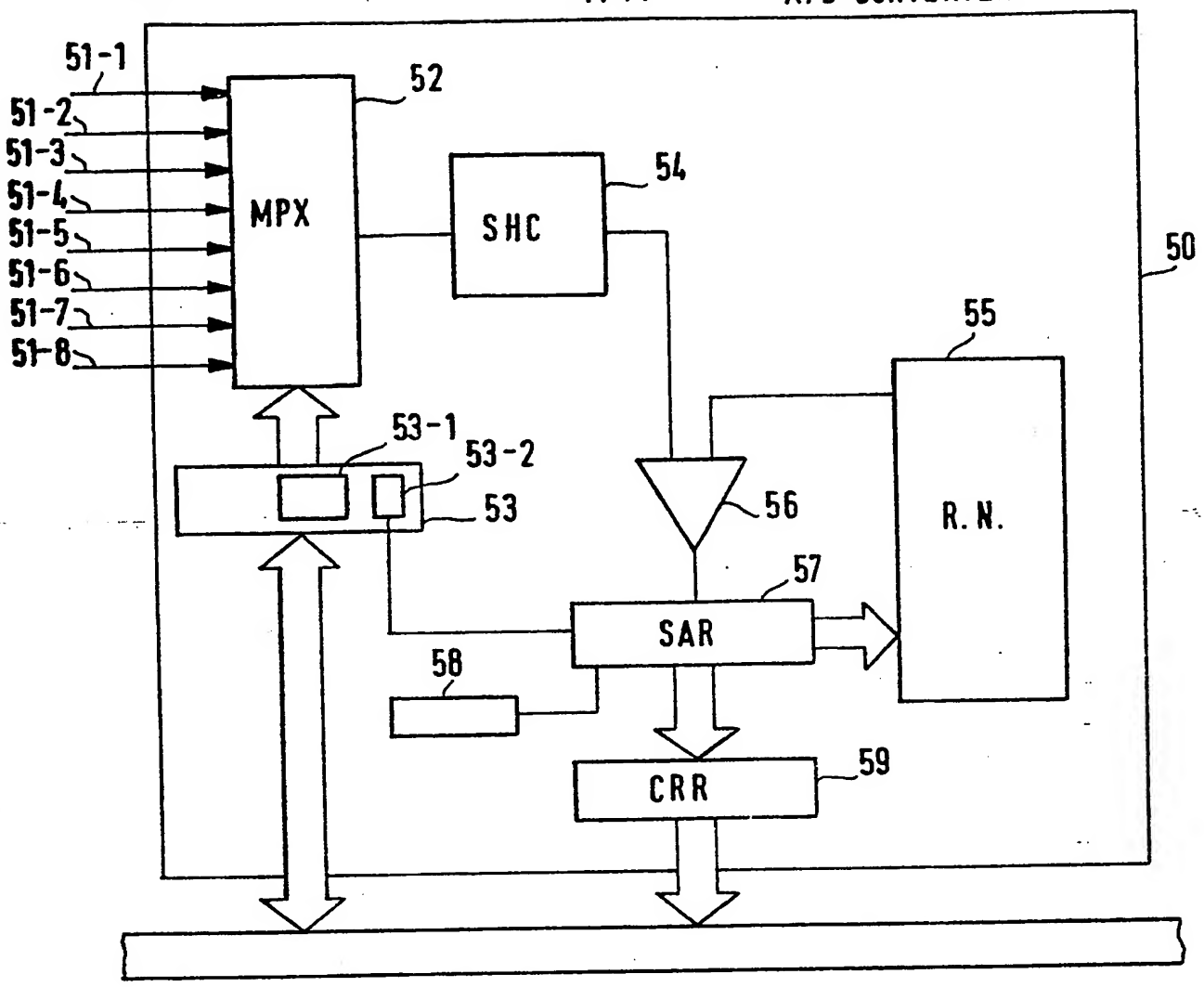

0 V
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
119'
120
121
122
200
210
300
310
30
31
32
33
34
35
36
R
S
Q

FIG. 4

FIG. 5

A/D CONVERTER



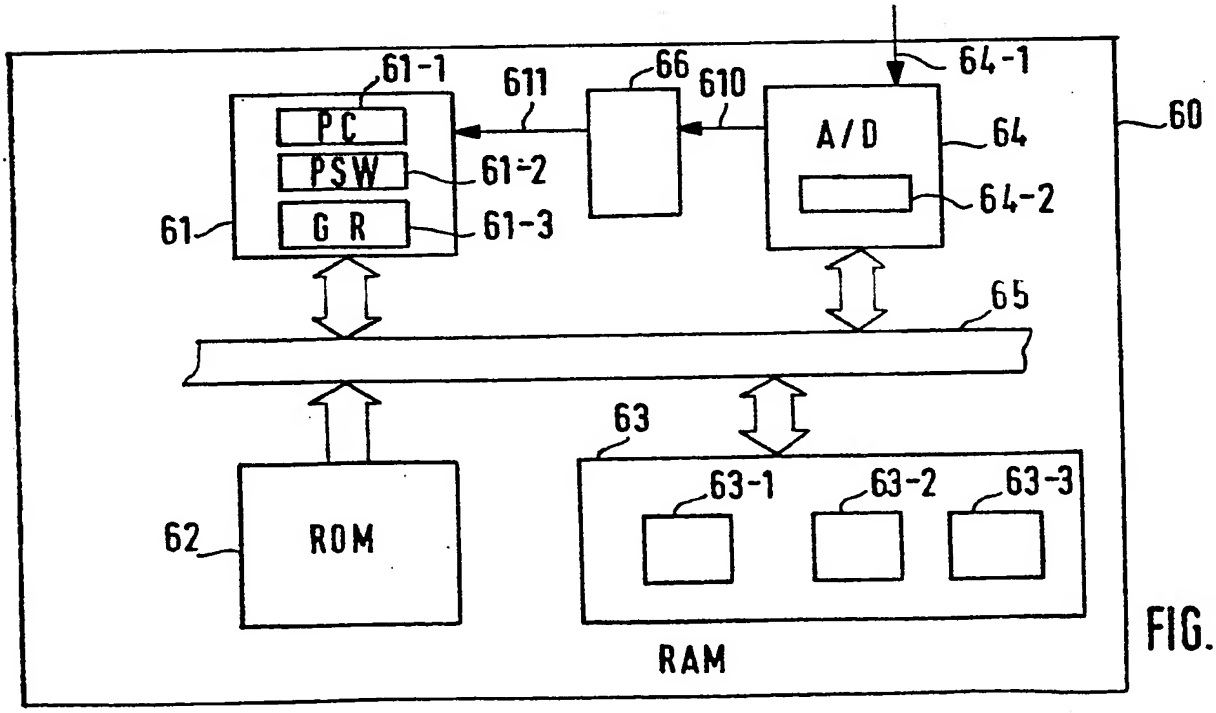


FIG. 6

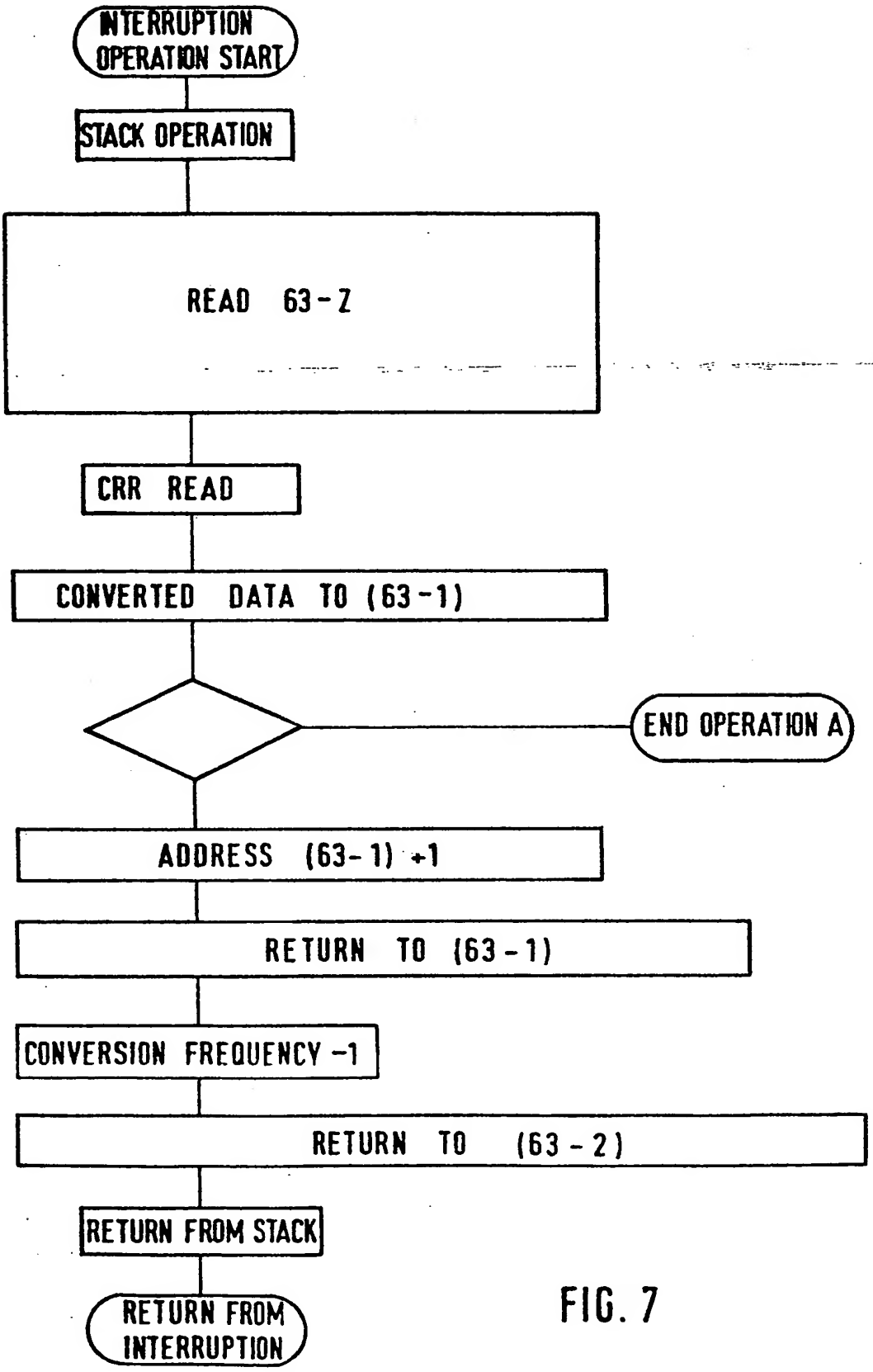

INTERRUPTION OPERATION START
STACK OPERATION
READ 63-Z
CRR READ
CONVERTED DATA TO (63-1)
END OPERATION A
ADDRESS (63-1) +1
RETURN TO (63-1)
CONVERSION FREQUENCY -1
RETURN TO (63-2)
RETURN FROM STACK
RETURN FROM INTERRUPTION

FIG. 7

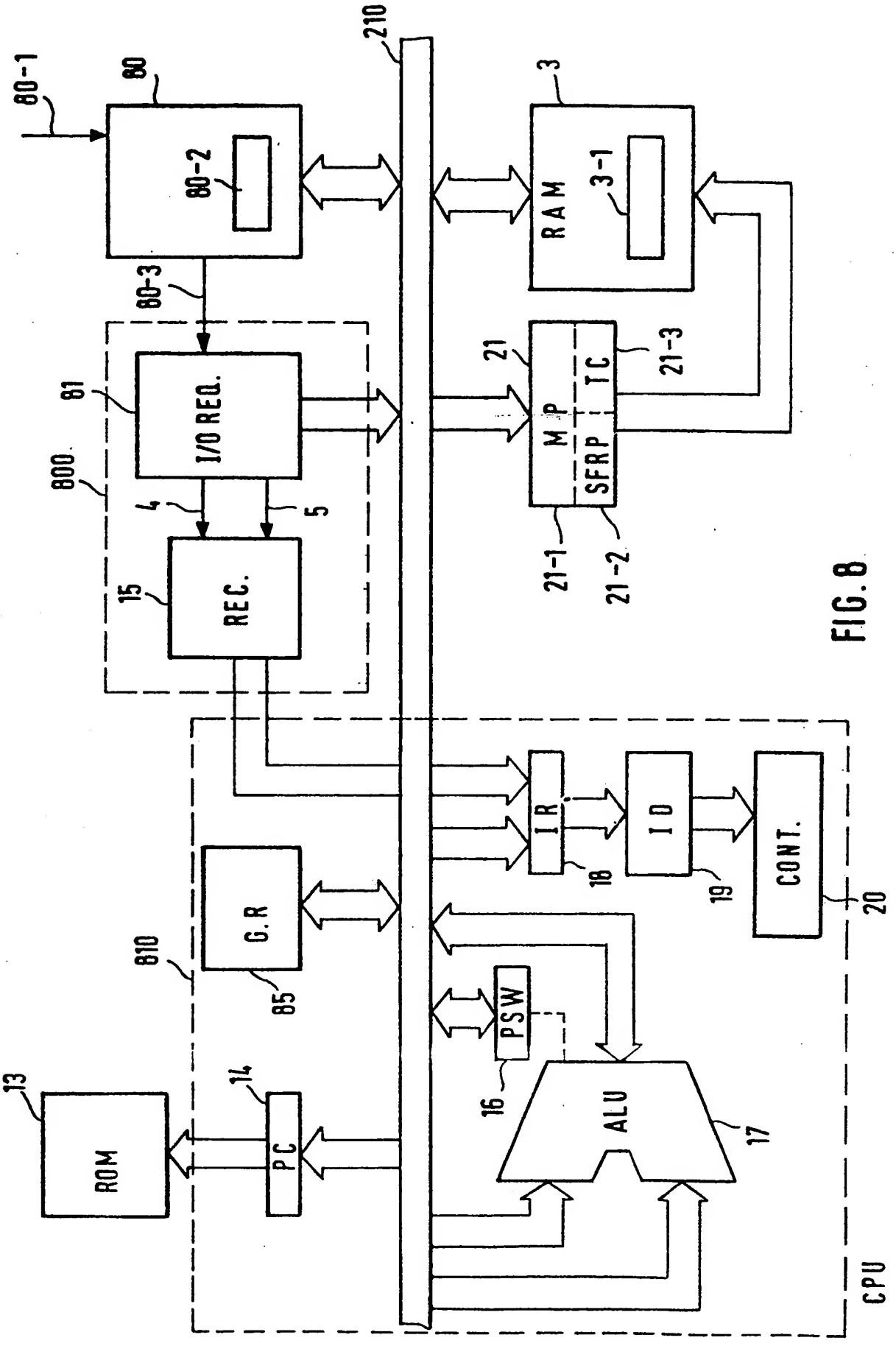

80-1
80
80-2
210
3
RAM
3-1
80-3
81
I/O REQ.
800
4
5
15
REC.
21
M P
TC
SFRP
21-3
21-1
21-2
I R
I D
CONT.
18
19
20
G.R
810
85
PSW
16
ALU
17
13
ROM
14
PC
CPU

FIG. 8

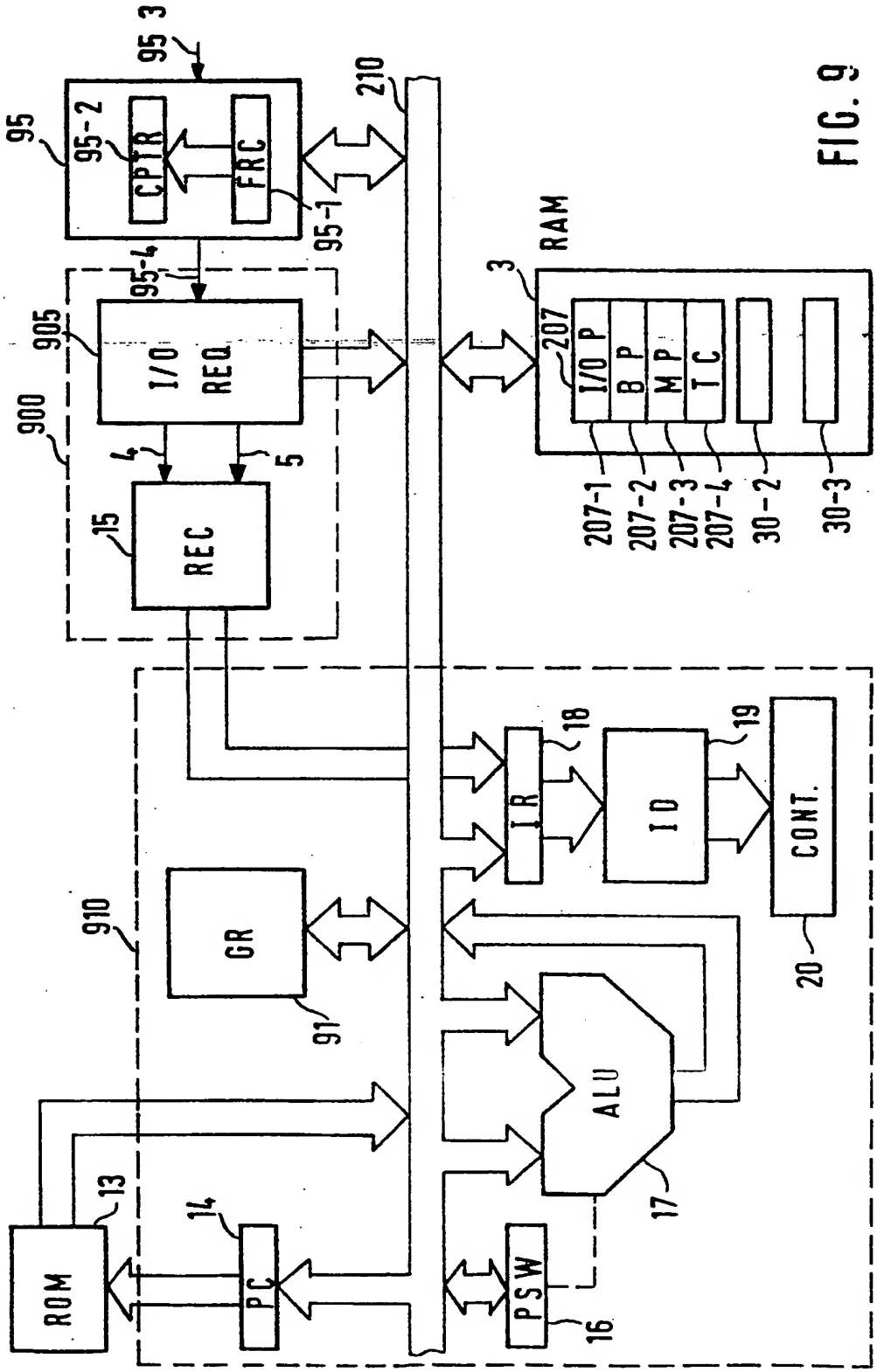

95 3
95
95-2
CPTR
FRC
95-1
210
95-4
905
I/O REQ
900
4
5
15
REC
3
RAM
207
I/O P
B P
M P
T C
207-1
207-2
207-3
207-4
30-2
30-3
18
IR
ID
19
CONT.
20
910
GR
91
ALU
17
13
ROM
14
PC
PSW
16
CPU

FIG. 9

FIG. 10

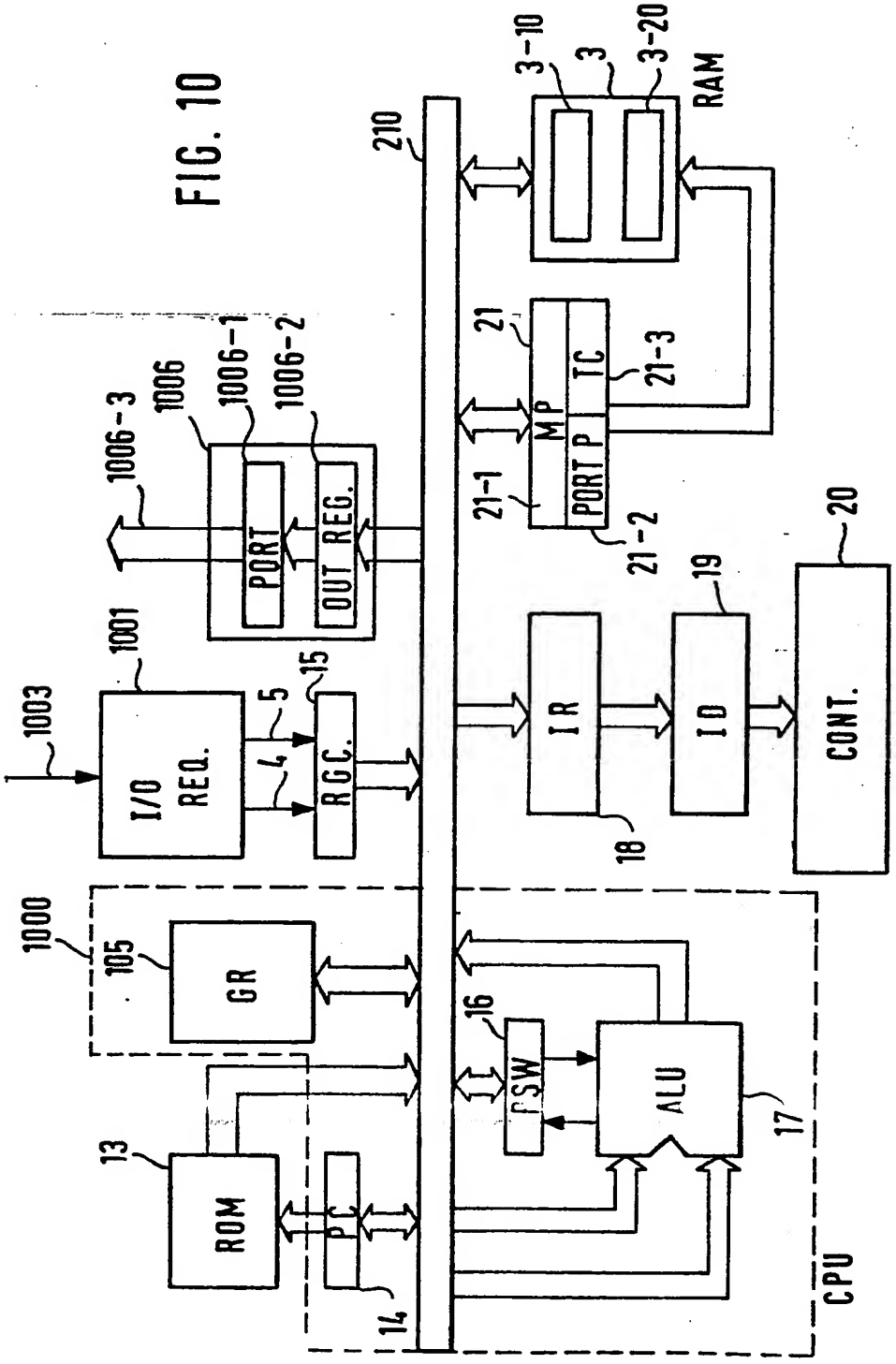
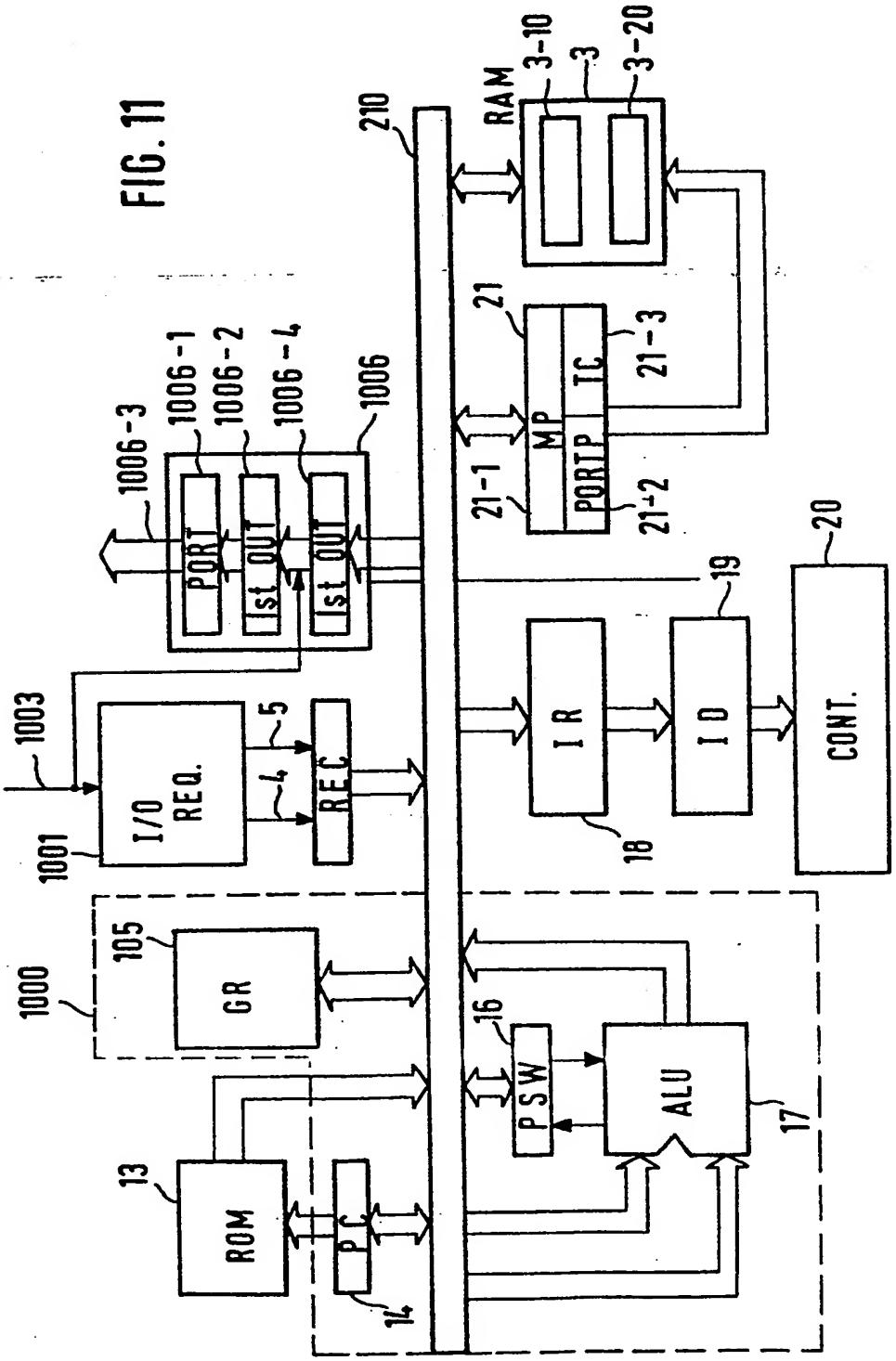

210
3-10
3
3-20
RAM
21
MP
TC
21-3
PORT P
21-1
21-2
1006
1006-1
1006-2
1006-3
PORT
OUT REG.
1001
1003
I/O REQ.
5
4
15
RGC.
I R
I D
CONT.
19
20
18
1000
105
GR
16
FSW
ALU
17
13
ROM
PC
14
CPU

FIG. 11


210
RAM
3-10
3
3-20
21
MP
IC
PORTP
21-1
21-2
21-3
1006-3
1006-1
1006-2
1006-4
1006
PORT
1st OUT
1st OUT
1003
1001
I/O REQ.
5
4
REC
I R
I O
CONT.
19
20
18
1000
105
GR
16
PSW
ALU
17
13
ROM
PC
14

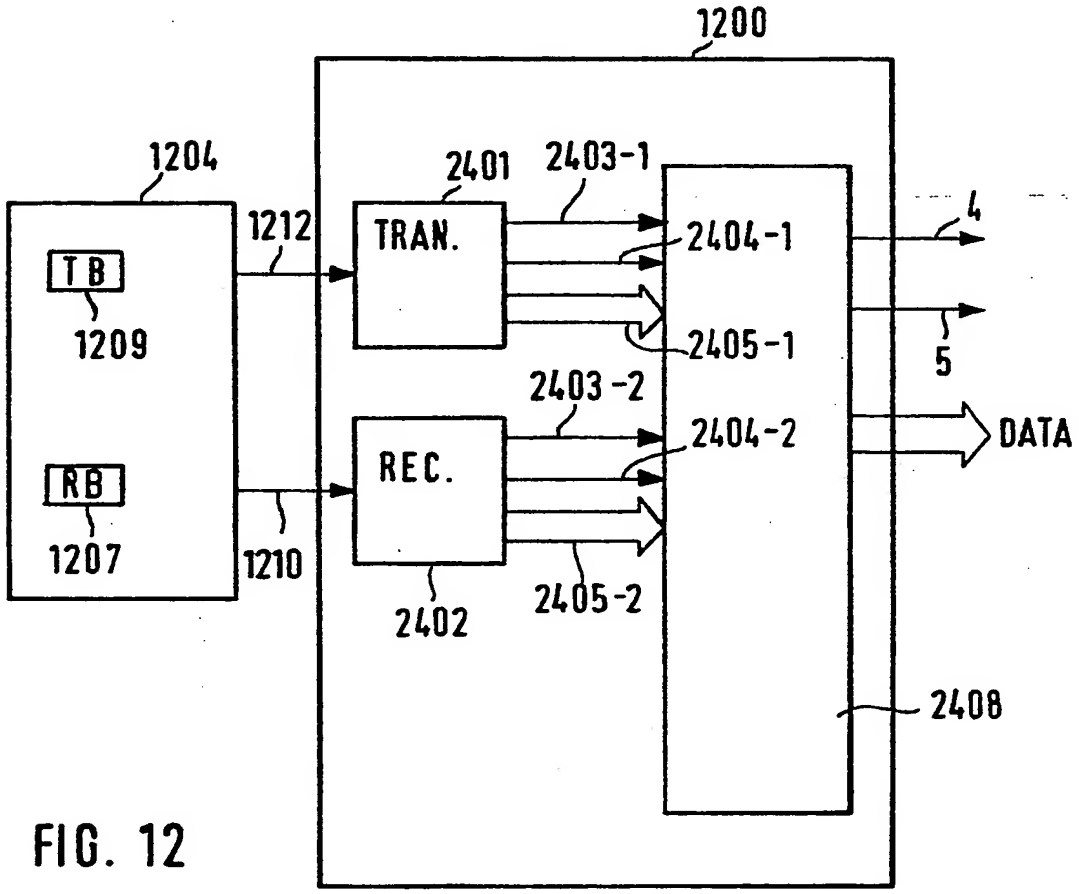

1200
1204
2401
2403-1
1212
TRAN.
2404-1
4
TB
1209
2405-1
5
2403-2
2404-2
DATA
REC.
RB
1207
1210
2405-2
2402
2408

FIG. 12

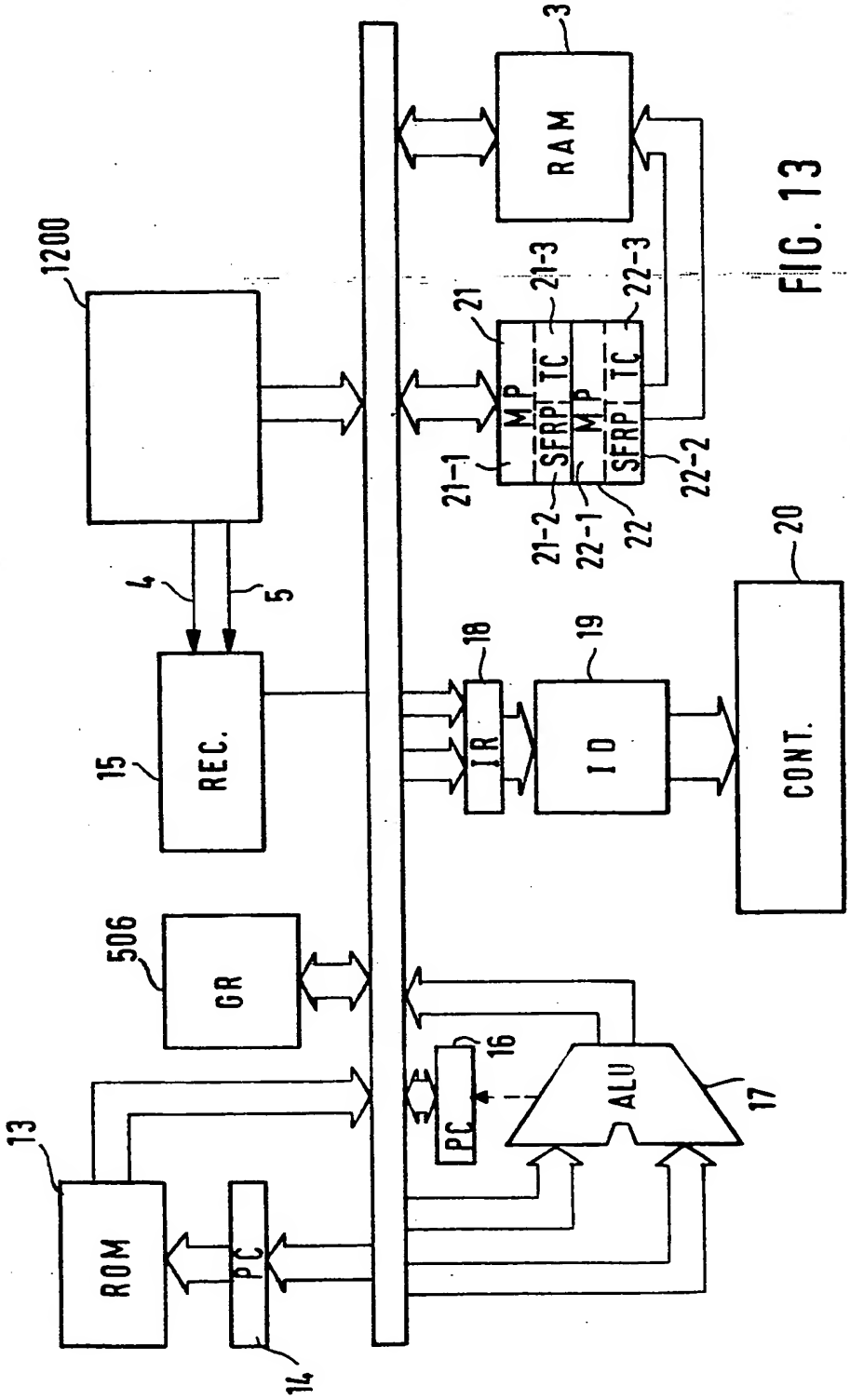

RAM
3
1200
21
21-3
22-3
MP
SFRP
TC
21-1
21-2
22-1
22
22-2
4
5
REC.
15
18
IR
19
ID
CONT.
20
506
GR
16
PC
ALU
17
13
ROM
PC
14

FIG. 13

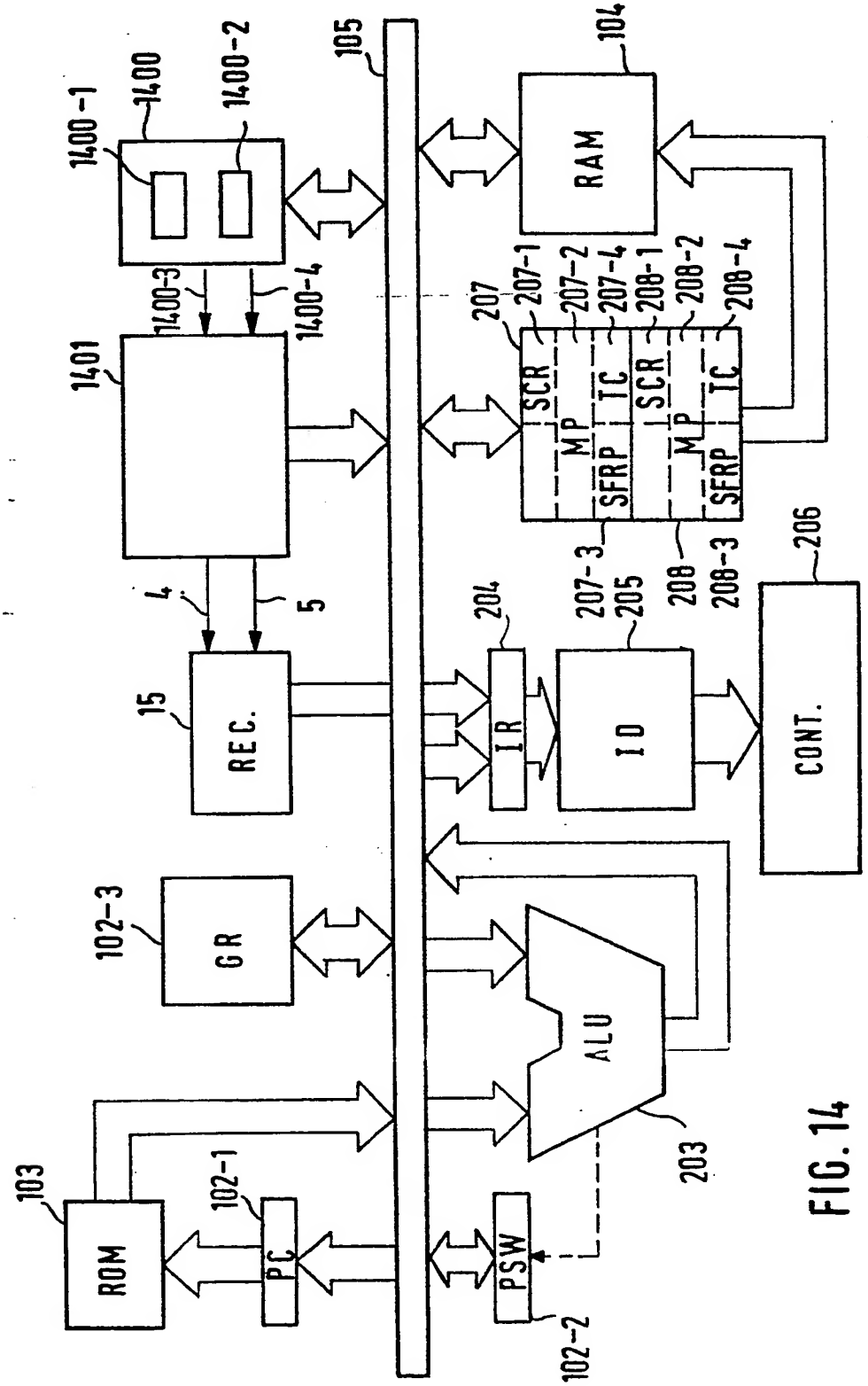

1400-1
1400
1400-2
105
104
RAM
1400-3
1400-4
207
207-1
207-2
207-4
208-1
208-2
208-4
1401
SCR
MP
SFRP
TC
SCR
MP
SFRP
TC
4
5
207-3
208
208-3
206
204
205
15
REC.
IR
ID
CONT.
102-3
GR
ALU
203
103
102-1
ROM
PC
PSW
102-2

FIG. 14

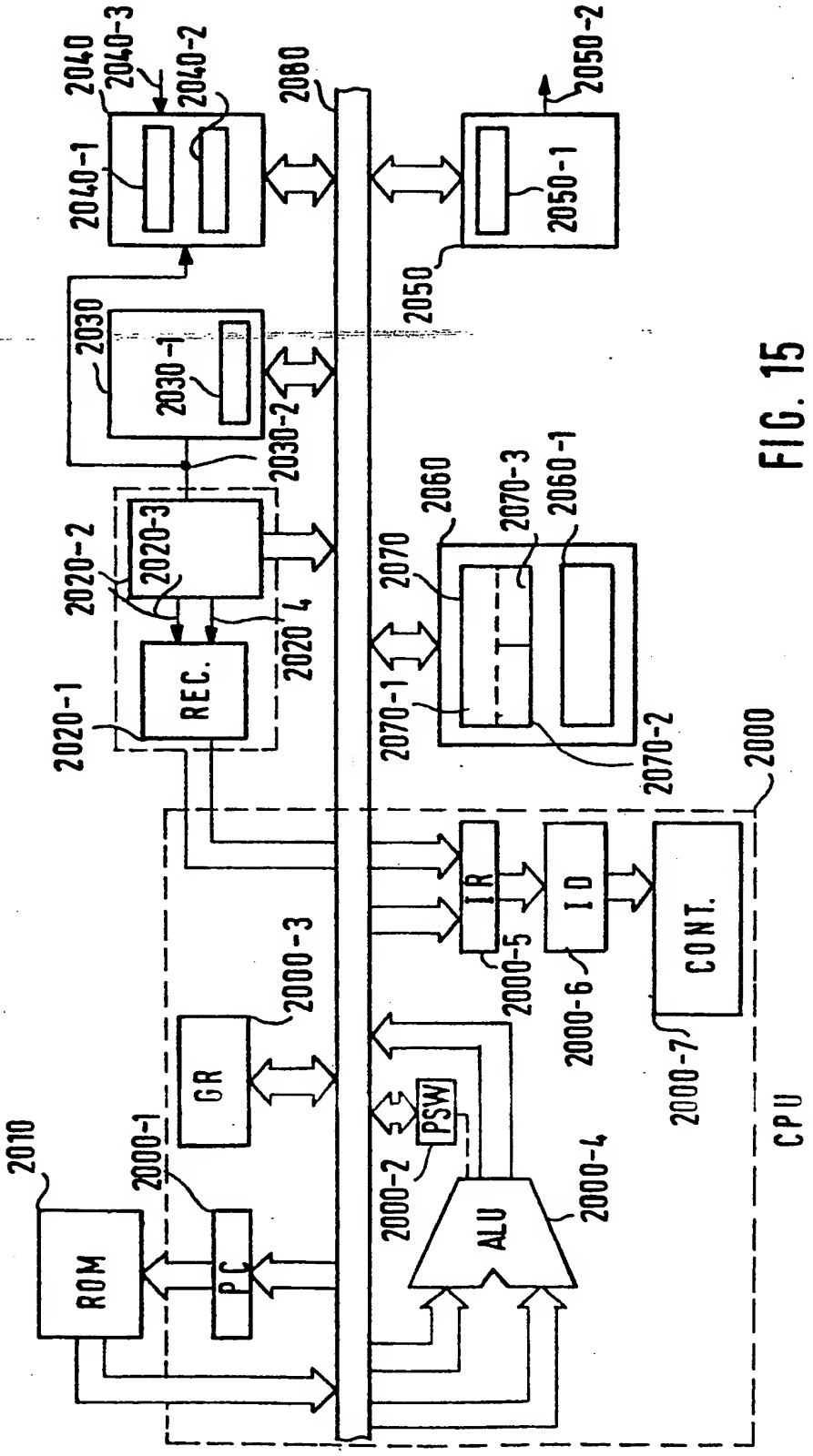

2040
2040-3
2040-2
2080
2050-2
2040-1
2050-1
2050
2030
2030-1
2030-2
2020-2
2020-3
2020-4
2060
2070-3
2060-1
2070
2020-1
REC.
2070-1
2070-2
2000
IR
ID
CONT.
2000-5
2000-6
2000-7
2000-3
GR
PSW
2000-2
2000-4
ALU
2010
2000-1
ROM
PC
CPU

FIG. 15

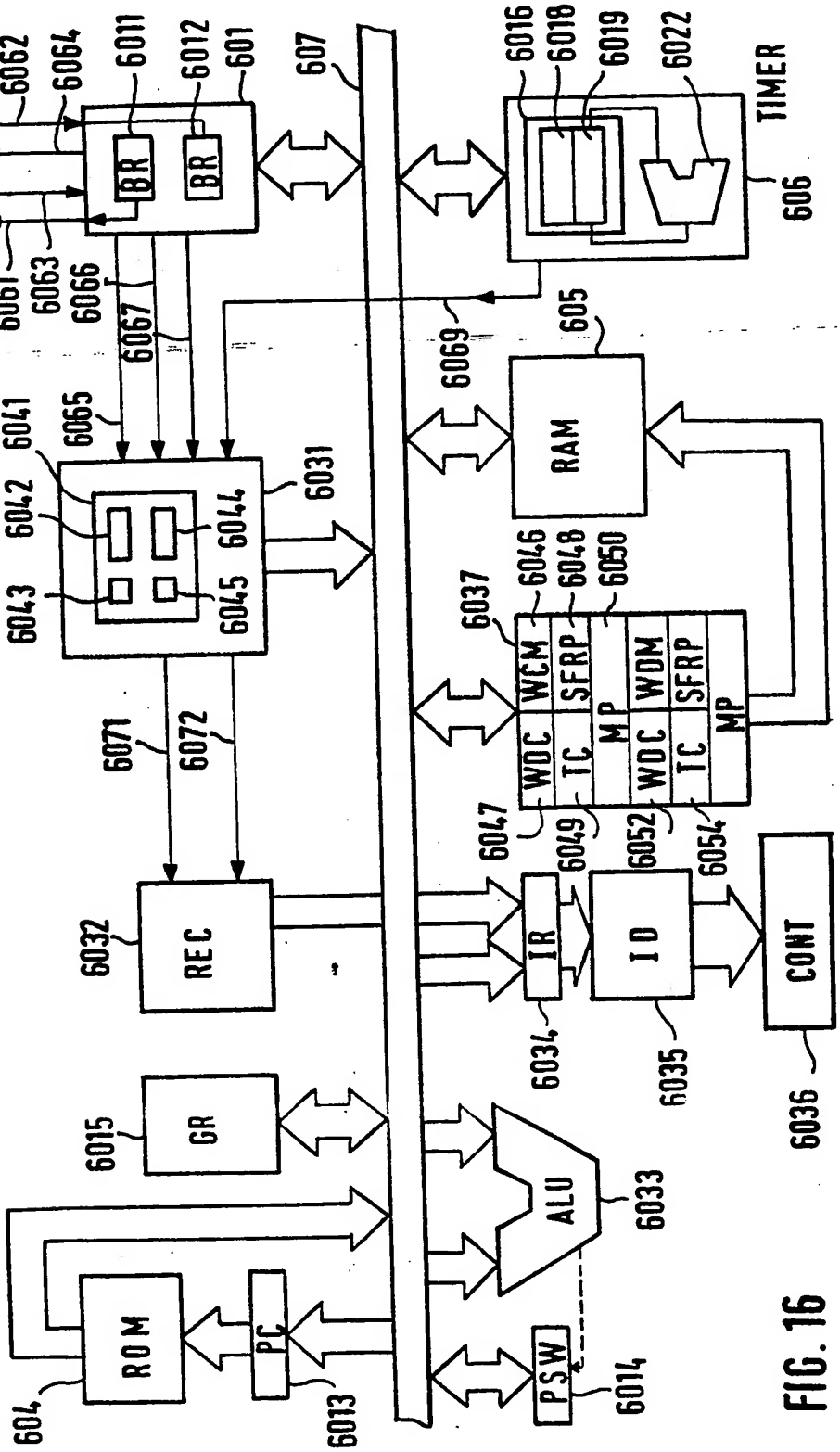

6062
6064
6011
6012
601
607
6016
6018
6019
6022
TIMER
BR
BR
606
6061
6063
6066
6067
6069
605
RAM
6041
6065
6031
6042
6044
6043
6045
6037
6046
6048
6050
WCM
SFRP
MP
WDM
SFRP
MP
WDC
TC
WDC
TC
6071
6072
6047
6049
6052
6054
6032
REC
IR
IO
CONT
6034
6035
6036
6015
GR
ALU
6033
ROM
PC
PSW
604
6013
6014

FIG. 16